

HANDOUTS

MODULE 1

Hard Disk

A hard disk is part of a unit, often called a "disk drive," "hard drive," or "hard disk drive," that stores and provides relatively quick access to large amounts of data on an electromagnetically charged surface or set of surfaces. Today's computers typically come with a hard disk that contains several billion bytes (gigabytes) of storage.

A hard disk is really a set of stacked "disks," each of which, like phonograph records, has data recorded electromagnetically in concentric circles or "tracks" on the disk. A "head" (something like a phonograph arm but in a relatively fixed position) records (writes) or reads the information on the tracks. Two heads, one on each side of a disk, read or write the data as the disk spins. Each read or write operation requires that data be located, which is an operation called a "seek." (Data already in a disk cache, however, will be located more quickly.)

A hard disk/drive unit comes with a set rotation speed varying from 4500 to 7200 rpm. Disk access time is measured in milliseconds. Although the physical location can be identified with cylinder, track, and sector locations, these are actually mapped to a logical block address (LBA) that works with the larger address range on today's hard disks.

Disk Cache

A disk cache is a mechanism for improving the time it takes to read from or write to a hard disk. Today, the disk cache is usually included as part of the hard disk. A disk cache can also be a specified portion of random access memory (RAM).

The disk cache holds data that has recently been read and, in some cases, adjacent data areas that are likely to be accessed next. Write caching is also provided with some disk caches.

History of Unix/Linux

1965 Bell Laboratories joins with MIT and General Electric in the development effort for the new operating system, Multics, which would provide multi-user, multi-processor, and multi-level (hierarchical) file system, among its many forward-looking features.

1969 AT&T was unhappy with the progress and drops out of the Multics project. Some of the Bell Labs programmers who had worked on this project, Ken Thompson, Dennis Ritchie, Rudd Canaday, and Doug McIlroy designed and implemented the first version of the Unix File System on a PDP-7 along with a few utilities. It was given the name UNIX by Brian Kernighan as a pun on Multics.

1971 The system now runs on a PDP-11, with 16Kbytes of memory, including 8Kbytes for user programs and a 512Kbyte disk.

Its first real use is as a text processing tool for the patent department at Bell Labs. That utilization justified further research and development by the programming group. UNIX caught on among programmers because it was designed with these features:

- programmers environment
- simple user interface
- simple utilities that can be combined to perform powerful functions
- hierarchical file system
- simple interface to devices consistent with file format
- multi-user, multi-process system
- architecture independent and transparent to the user.

1973 Unix is re-written mostly in C, a new language developed by Dennis Ritchie. Being written in this high-level language greatly decreased the effort needed to port it to new machines.

1974 Thompson and Ritchie publish a paper in the Communications of the ACM describing the new Unix OS. This generates enthusiasm in the Academic community which sees a potentially great teaching tool for studying programming systems development. Since AT&T is prevented from marketing the product due to the 1956 Consent Decree they license it to Universities for educational purposes and to commercial entities.

1977 There are now about 500 Unix sites world-wide.

1980 BSD 4.1 (Berkeley Software Development)

1983 SunOS, BSD 4.2, SysV

1984 There are now about 100,000 Unix sites running on many different hardware platforms, of vastly different capabilities.

1988 AT&T and Sun Microsystems jointly develop System V Release 4 (SVR4). This would later be developed into UnixWare and Solaris 2.

1993 Novell buys UNIX from AT&T

1994 Novell gives the name "UNIX" to X/OPEN

1995 Santa Cruz Operations buys UnixWare from Novell. Santa Cruz Operations and Hewlett-Packard announce that they will jointly develop a 64-bit version of Unix.

1996 International Data Corporation forecasts that in 1997 there will be 3 million Unix systems shipped world-wide.

Inside Linux

- **Kernel**
 - The core of the UNIX system. Loaded at system start up (boot). Memory-resident control program.
 - Manages the entire resources of the system, presenting them to you and every other user as a coherent system. Provides service to user applications such as device management, process scheduling, etc.
 - Example functions performed by the kernel are:
 - Managing the machine's memory and allocating it to each process.
 - Scheduling the work done by the CPU so that the work of each user is carried out as efficiently as is possible.
 - Accomplishing the transfer of data from one part of the machine to another
 - Interpreting and executing instructions from the shell
 - Enforcing file access permissions
 - You do not need to know anything about the kernel in order to use a UNIX system. These details are provided for your information only.
- **Shell**
 - Whenever you login to a Unix system you are placed in a shell program. The shell's prompt is usually visible at the cursor's position on your screen. To get your work done, you enter commands at this prompt.
 - The shell is a command interpreter; it takes each command and passes it to the operating system kernel to be acted upon. It then displays the results of this operation on your screen.
 - Several shells are usually available on any UNIX system, each with its own strengths and weaknesses.
 - Different users may use different shells. Initially, your system administrator will supply a default shell, which can be overridden or changed. The most commonly available shells are:
 - Bourne shell (sh)
 - C shell (csh)
 - Korn shell (ksh)
 - TC Shell (tcsh)
 - Bourne Again Shell (bash)

- Each shell also includes its own programming language. Command files, called "shell scripts" are used to accomplish a series of tasks.
- **Utilities**
 - UNIX provides several hundred utility programs, often referred to as commands.
 - Accomplish universal functions
 - editing
 - file maintenance
 - printing
 - sorting
 - programming support
 - online info etc.
 - Modular: single functions can be grouped to perform more complex tasks

Operating system

An operating system or OS is a software program that enables the computer hardware to communicate and operate with the computer software. Without a computer operating system, a computer and software programs would be useless.

An operating system (sometimes abbreviated as "OS") is the program that, after being initially loaded into the computer by a boot program, manages all the other programs in a computer. The other programs are called applications or application programs. The application programs make use of the operating system by making requests for services through a defined application program interface (API). In addition, users can interact directly with the operating system through a user interface such as a command language or a graphical user interface (GUI).

An operating system performs these services for applications:

- In a multitasking operating system where multiple programs can be running at the same time, the operating system determines which applications should run in what order and how much time should be allowed for each application before giving another application a turn.
- It manages the sharing of internal memory among multiple applications.
- It handles input and output to and from attached hardware devices, such as hard disks, printers, and dial-up ports.
- It sends messages to each application or interactive user (or to a system operator) about the status of operation and any errors that may have occurred.
- It can offload the management of what are called batch jobs (for example, printing) so that the initiating application is freed from this work.
- On computers that can provide parallel processing, an operating system can manage how to divide the program so that it runs on more than one processor at a time.

Examples of computer operating systems

- Redhat – Very popular Linux operating system from Redhat
- Microsoft Windows - PC and IBM compatible operating system. Microsoft Windows is the most commonly found and used operating system in PCs
- Apple MacOS - Apple computer operating system. The only Apple computer operating system.
- Ubuntu Linux - A popular variant of Linux used with PC and IBM compatible computers.
- Google Android - operating system used with Android compatible phones.
- iOS - Operating system used with the Apple iPhone.

Various Parts of an Operating System

UNIX and 'UNIX-like' operating systems (such as Linux) consist of a kernel and some system programs. There are also some application programs for doing work. The kernel is the heart of the operating system. In fact, it is often mistakenly considered to be the operating system itself, but it is not. An operating system provides many more services than a plain kernel.

It keeps track of files on the disk, starts programs and runs them concurrently, assigns memory and other resources to various processes, receives packets from and sends packets to the network, and so on. The kernel does very little by itself, but it provides tools with which all services can be built. It also prevents anyone from accessing the hardware directly, forcing everyone to use the tools it provides. This way the kernel provides some protection for users from each other. The tools provided by the kernel are used via system calls.

The system programs use the tools provided by the kernel to implement the various services required from an operating system. System programs, and all other programs, run 'on top of the kernel', in what is called the user mode. The difference between system and application programs is one of intent: applications are intended for getting useful things done (or for playing, if it happens to be a game), whereas system programs are needed to get the system working. A word processor is an application; mount is a system program. The difference is often somewhat blurry, however, and is important only to compulsive categorizers.

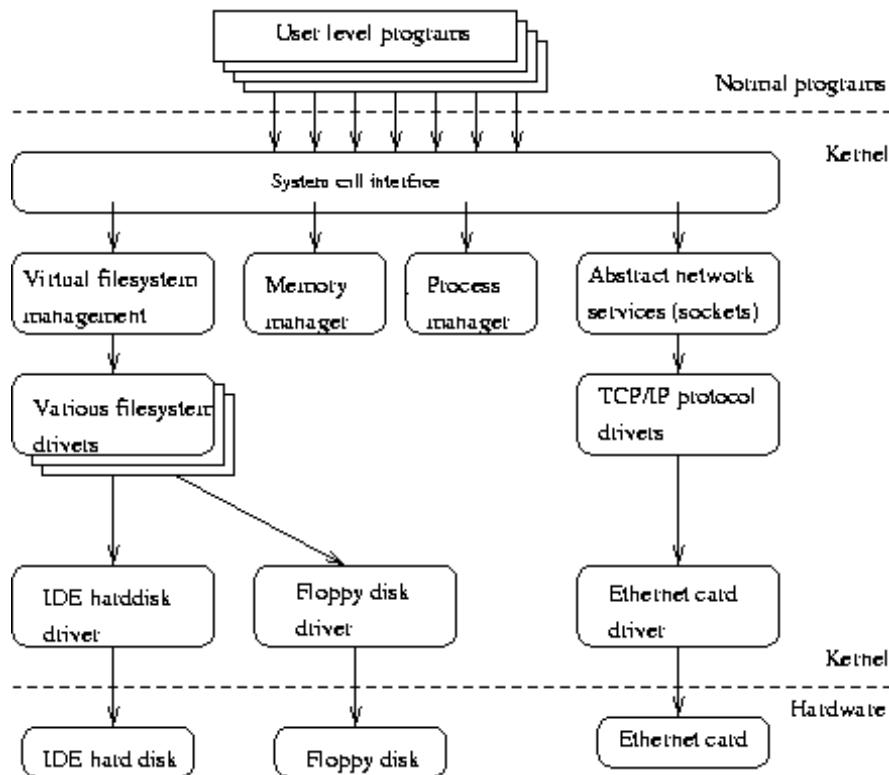
An operating system can also contain compilers and their corresponding libraries (GCC and the C library in particular under Linux), although not all programming languages need be part of the operating system. Documentation, and sometimes even games, can also be part of it.

Important parts of the kernel

The Linux kernel consists of several important parts:

- Process management
- Memory management
- Hardware device drivers
- Filesystem drivers
- Network management
- Various other bits and pieces

The following figure shows some of the more important parts of the Linux kernel



Probably the most important parts of the kernel (nothing else works without them) are memory management and process management. Memory management takes care of assigning memory areas and swap space areas to processes, parts of the kernel, and for the buffer cache. Process management creates processes, and implements multitasking by switching the active process on the processor.

At the lowest level, the kernel contains a hardware device driver for each kind of hardware it supports. Since the world is full of different kinds of hardware, the number of hardware device drivers is large. There are often many otherwise similar pieces of hardware that differ in how they are controlled by software. The similarities make it possible to have general classes of drivers that support similar operations; each member of the class has the same interface to the rest of the kernel but differs in what it needs to do to implement them. For example, all disk drivers look alike to the rest of the kernel, i.e., they all have operations like 'initialize the drive', 'read sector N', and 'write sector N'.

What is virtual memory?

Linux supports virtual memory, that is, using a disk as an extension of RAM so that the effective size of usable memory grows correspondingly. The kernel will write the contents of a currently unused block of memory to the hard disk so that the memory can be used for another purpose. When the original contents are needed again, they are read back into memory. This is all made completely transparent to the user; programs running under Linux only see the larger amount of memory available and don't notice that parts of them reside on the disk from time to time. Of course, reading and writing the hard disk is slower (on the order of a thousand times slower) than using real memory, so the programs don't run as fast. The part of the hard disk that is used as virtual memory is called the swap space.

Linux can use either a normal file in the filesystem or a separate partition for swap space. A swap partition is faster, but it is easier to change the size of a swap file (there's no need to repartition the whole hard disk, and possibly install everything from scratch). When you know how much swap space you need, you should go for a swap partition, but if you are uncertain, you can use a swap file first, use the system for a while so that you can get a feel for how much swap you need, and then make a swap partition when you're confident about its size.

You should also know that Linux allows one to use several swap partitions and/or swap files at the same time. This means that if you only occasionally need an unusual amount of swap space, you can set up an extra swap file at such times, instead of keeping the whole amount allocated all the time.

A note on operating system terminology: computer science usually distinguishes between swapping (writing the whole process out to swap space) and paging (writing only fixed size parts, usually a few kilobytes, at a time). Paging is usually more efficient, and that's what Linux does, but traditional Linux terminology talks about swapping anyway.

HANDOUTS

MODULE 2

CentOS 7 Installation Steps with Screenshots

CentOS community has released its Latest Operating System named as **CentOS 7**. Some of the **new features** in CentOS 7 as compared with CentOS 6.X are listed below :

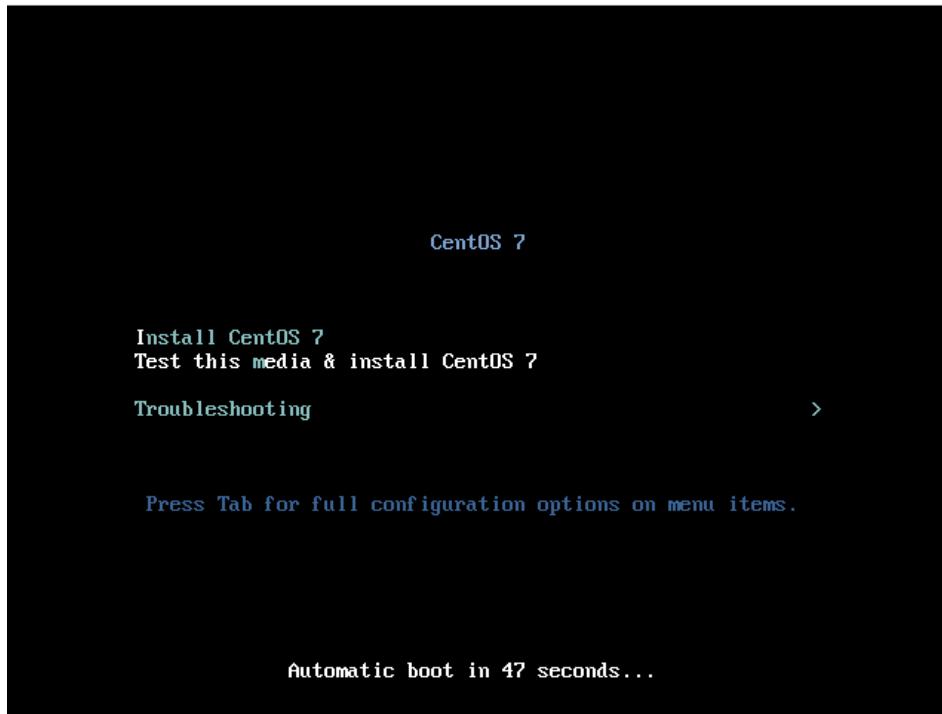
- CentOS 7 uses **XFS** as its default file system.
- **OpenJDK-7** is the default JDK.
- initd has been replaced by **systemd**.
- New Linux **Kernel 3.10.0**, support for **Linux Containers**, and the inclusion of the Open VMware Tools and 3D graphics drivers out of the box.
- Change in the numbering Scheme, Official release is **Centos 7.0-1406** , where as 7 Comes from RHEL 7 and 1406 shows release date(June 2014).

In this article we will go through the CentOS 7 Installations steps with screenshots.

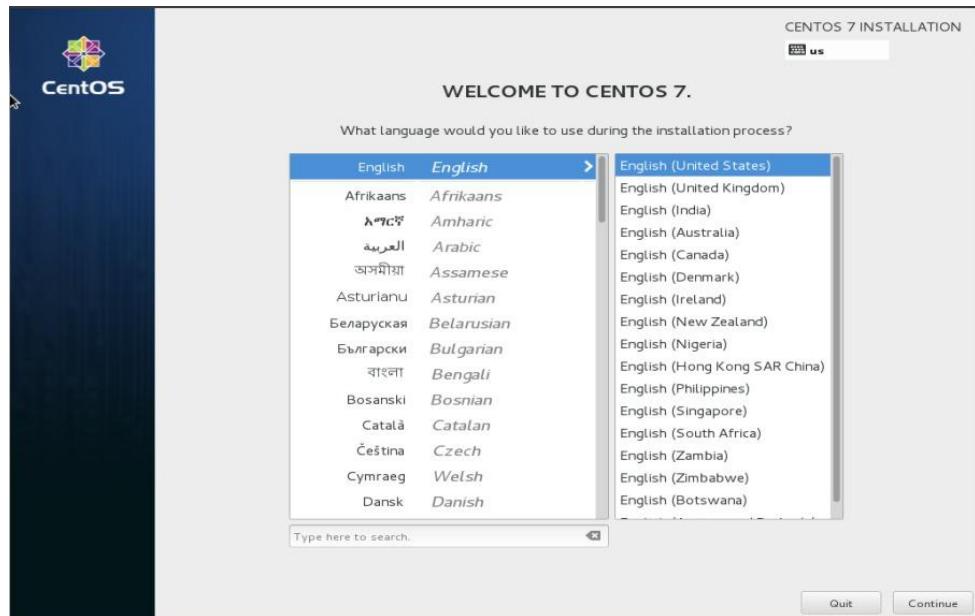
Step:1 First Download the **.iso** file from the **CentOS website**, burn it onto the disc. Boot your PC from DVD.

Use this link to [Download Centos 7](#) (64 bit)

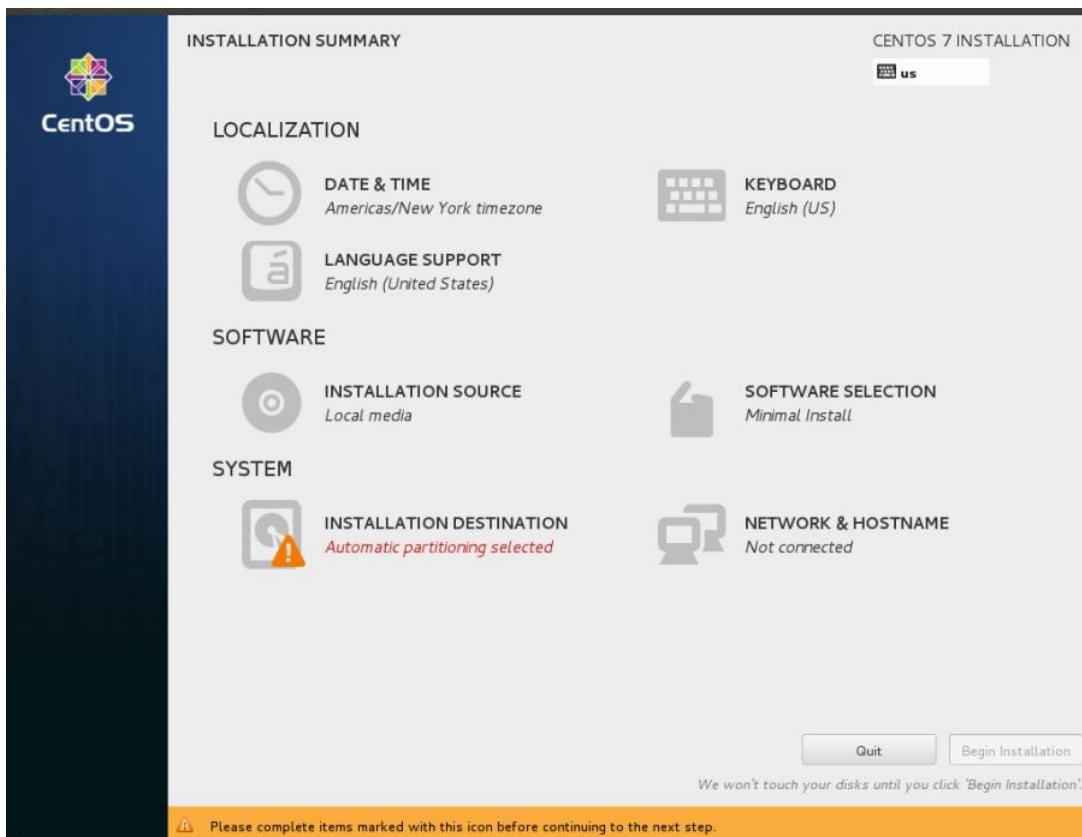
Step:2 Choose '**Install CentOS 7**' option and press enter



Step:3 Choose your respective Language and click on **Continue**, in my case i have choose English , as shown below



Step: 4 Change the **Installation Destination** , by default installer will do automatic partitioning for your hard disk. To create your own customize partition table click on '**Installation Destination**'.



As you can see below i have around 30 GB hard drive for OS installation. choose '**I will configure partitioning**' then click **Done**

INSTALLATION DESTINATION CENTOS 7 INSTALLATION

Done 

Device Selection

Select the device(s) you'd like to install to. They will be left untouched until you click on the main menu's "Begin Installation" button.

Local Standard Disks

29.87 GB

ATA VBOX HARDDISK
sda / 29.87 GB free

Disks left unselected here will not be touched.

Specialized & Network Disks

Add a disk...

Disks left unselected here will not be touched.

Other Storage Options

Partitioning

Automatically configure partitioning. I will configure partitioning.

I would like to make additional space available.

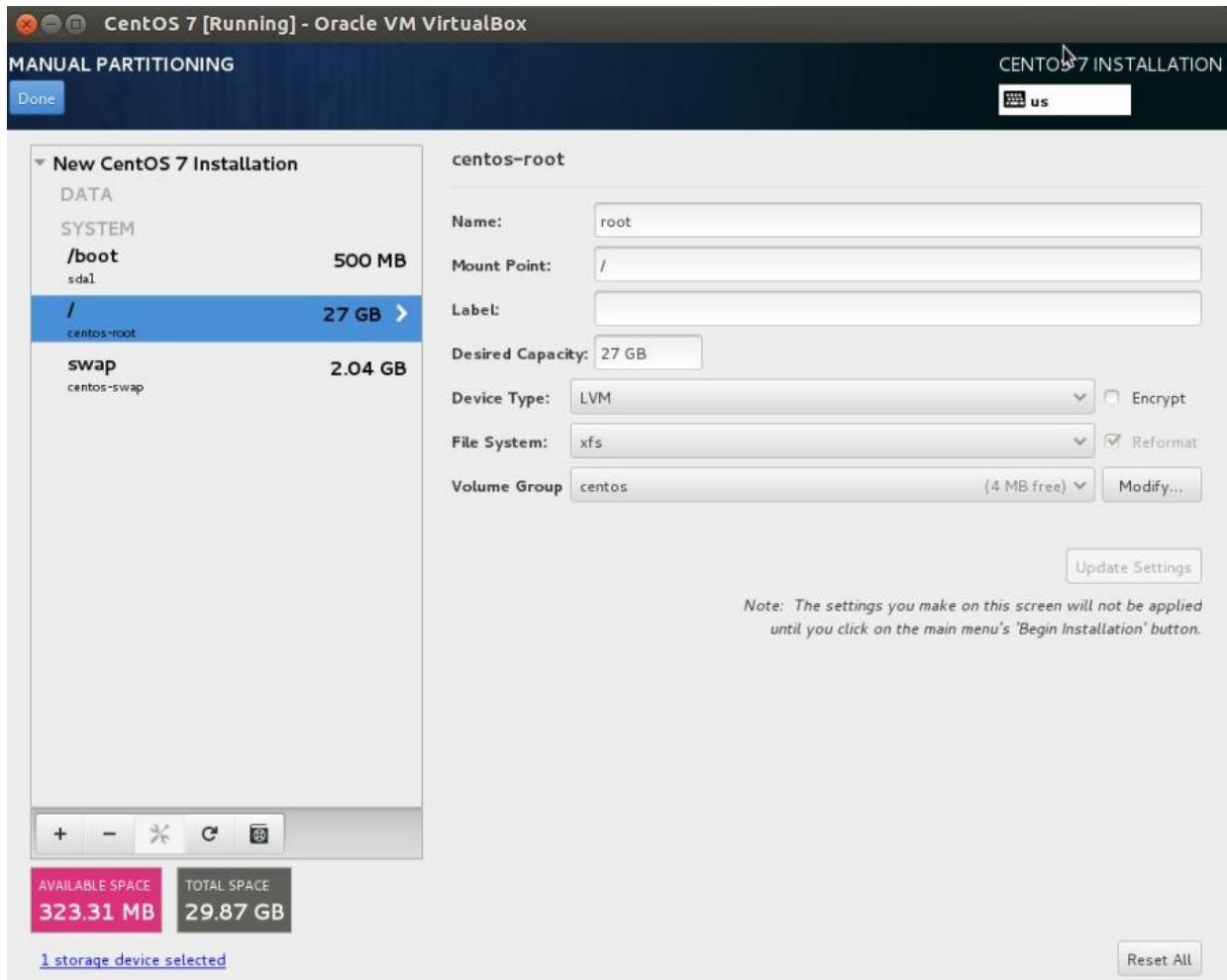
Encryption

Encrypt my data. *You'll set a passphrase later.*

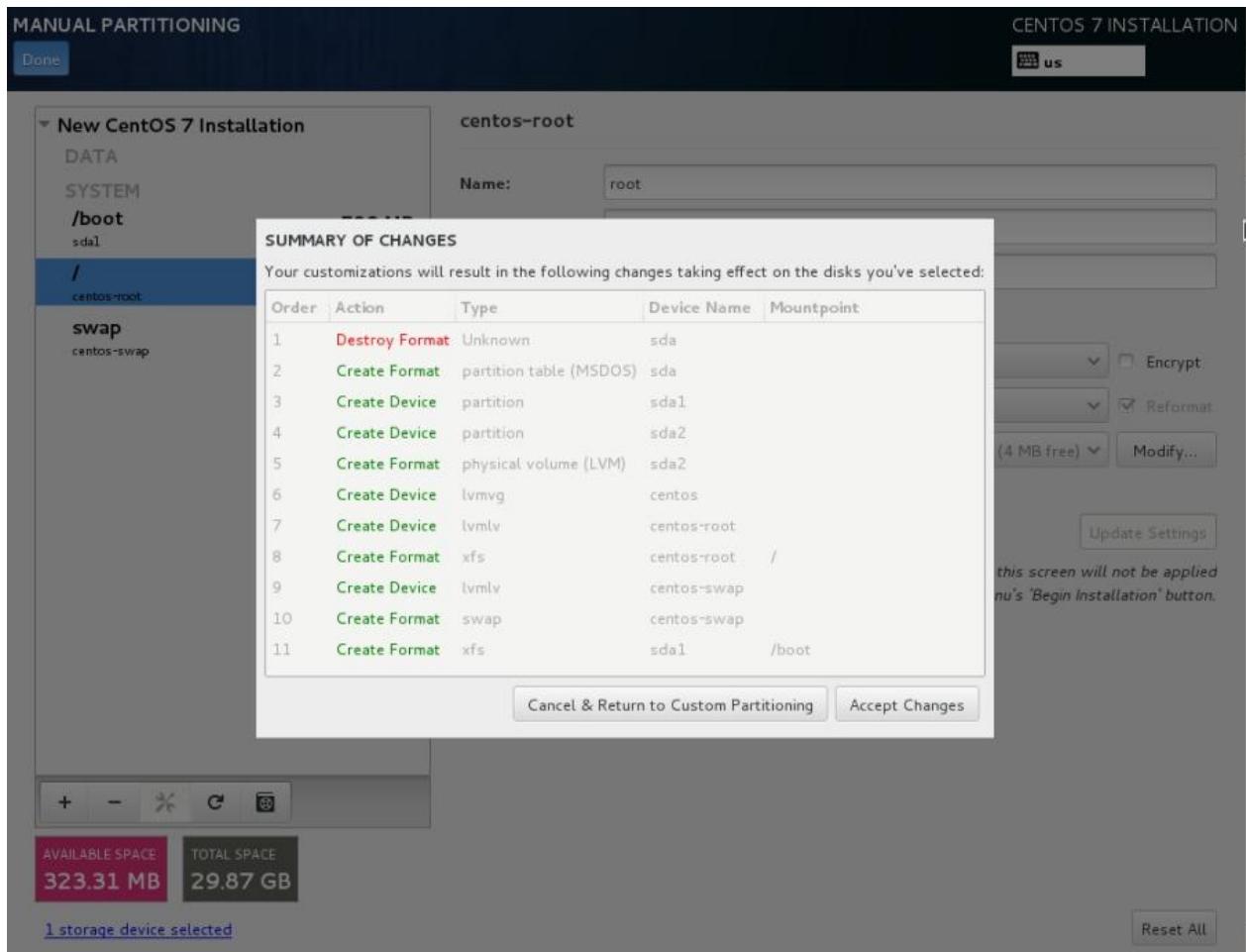
[Full disk summary and bootloader...](#)

1 disk selected; 29.87 GB capacity; 29.87 GB free

Step:5 Create the partition table , In my case i am putting everything under LVM and created **/boot** , / and **swap** partition as shown below :

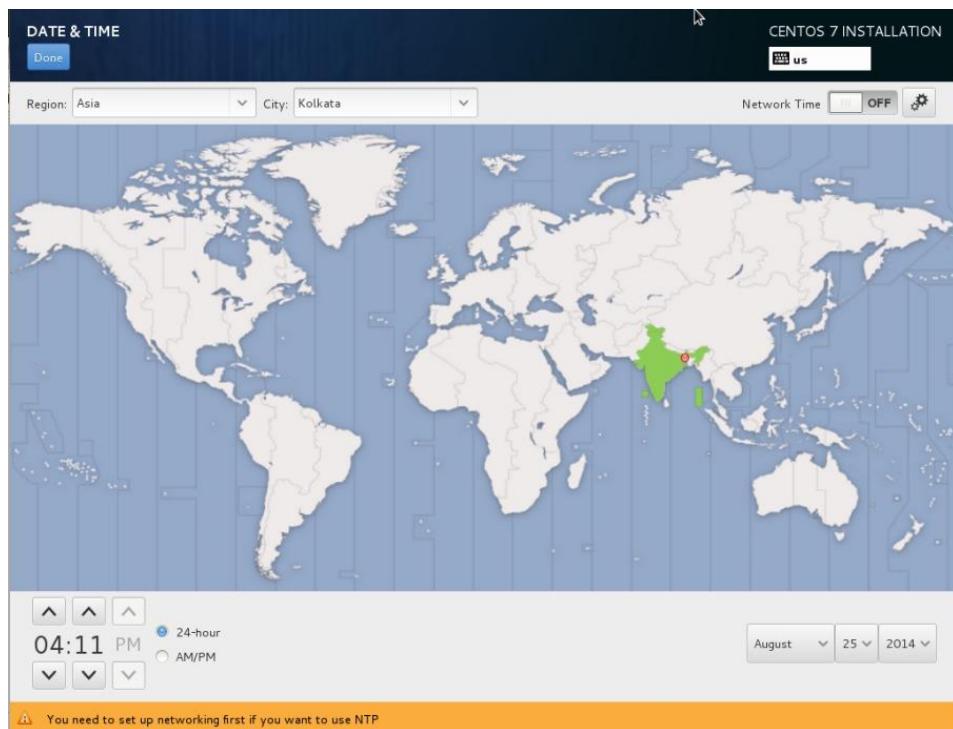


Click on Done

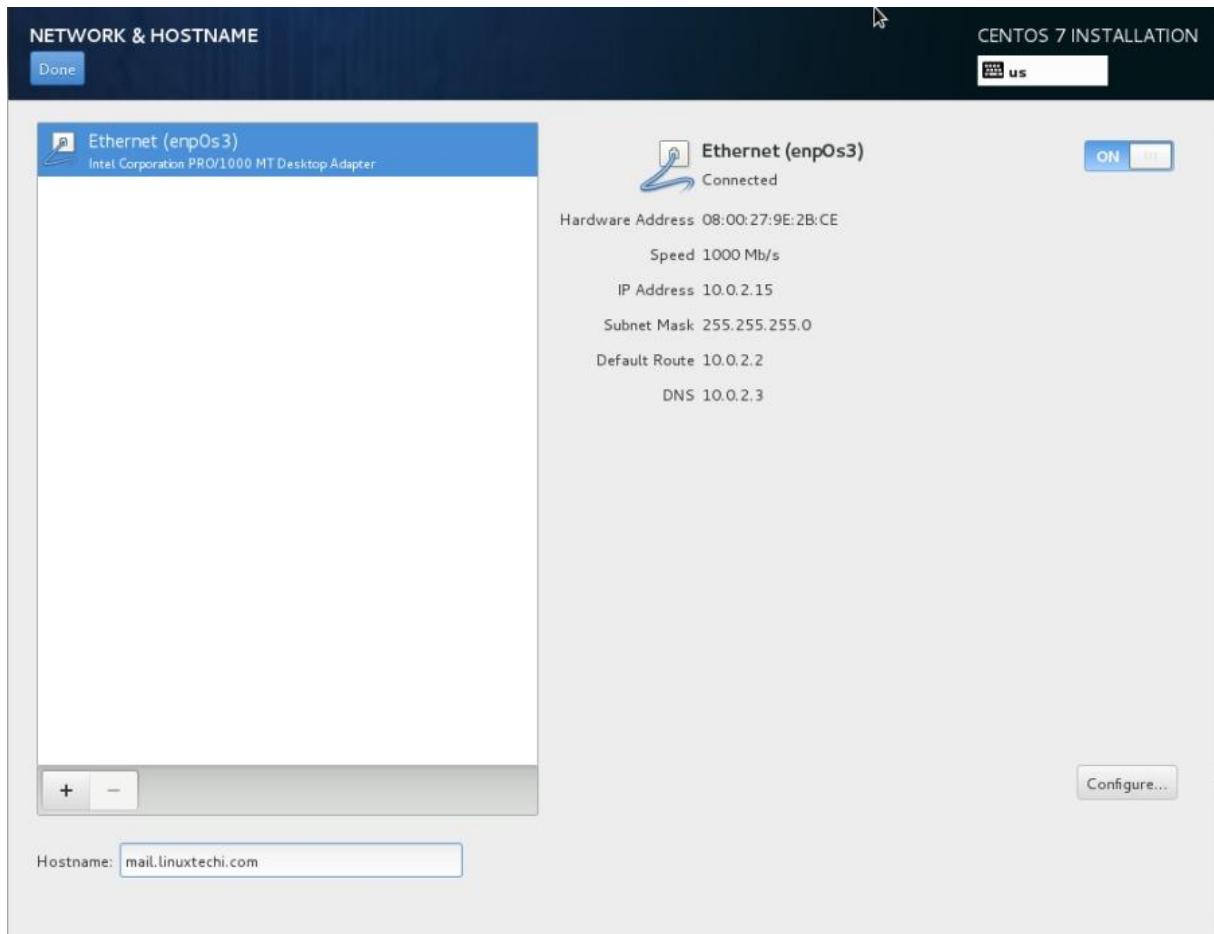


Click on Accept Changes

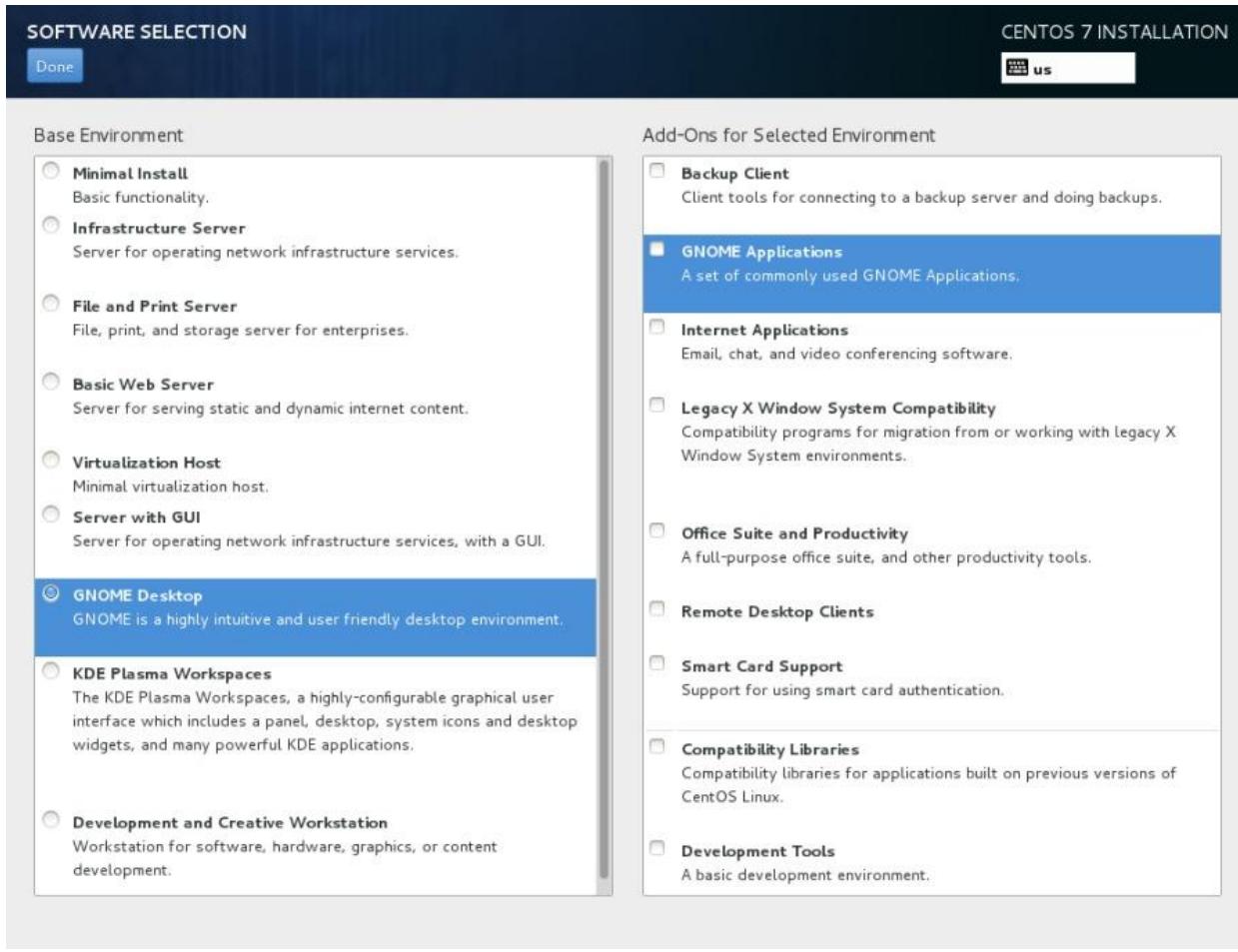
Step:6 Set Date & Time with your respective Zone



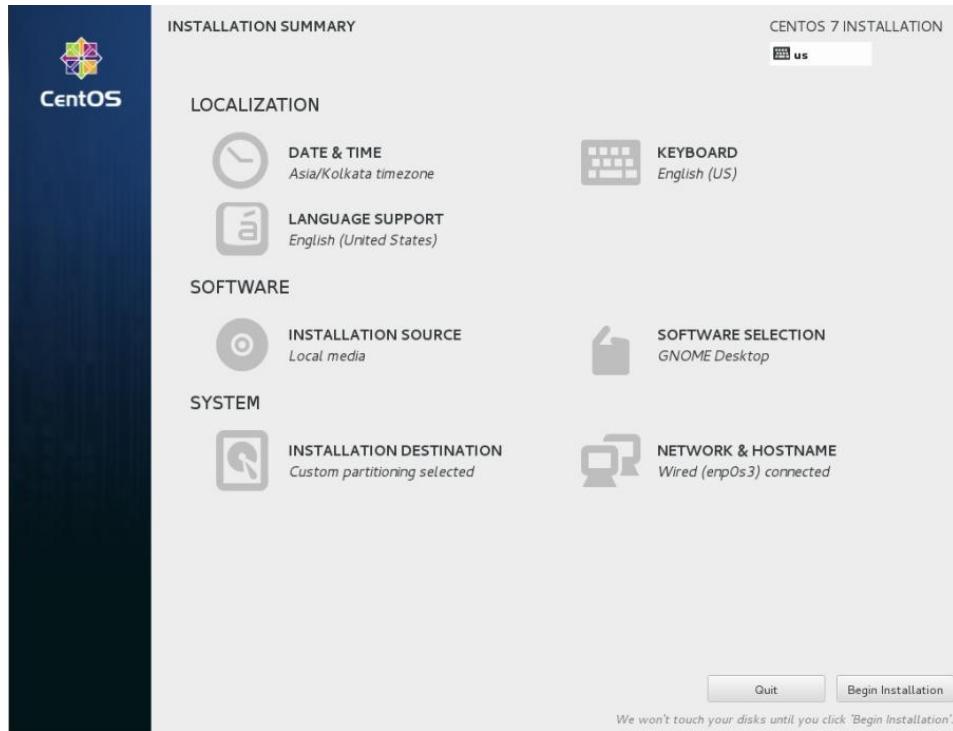
Step:7 Configure Networking and Set the hostname .



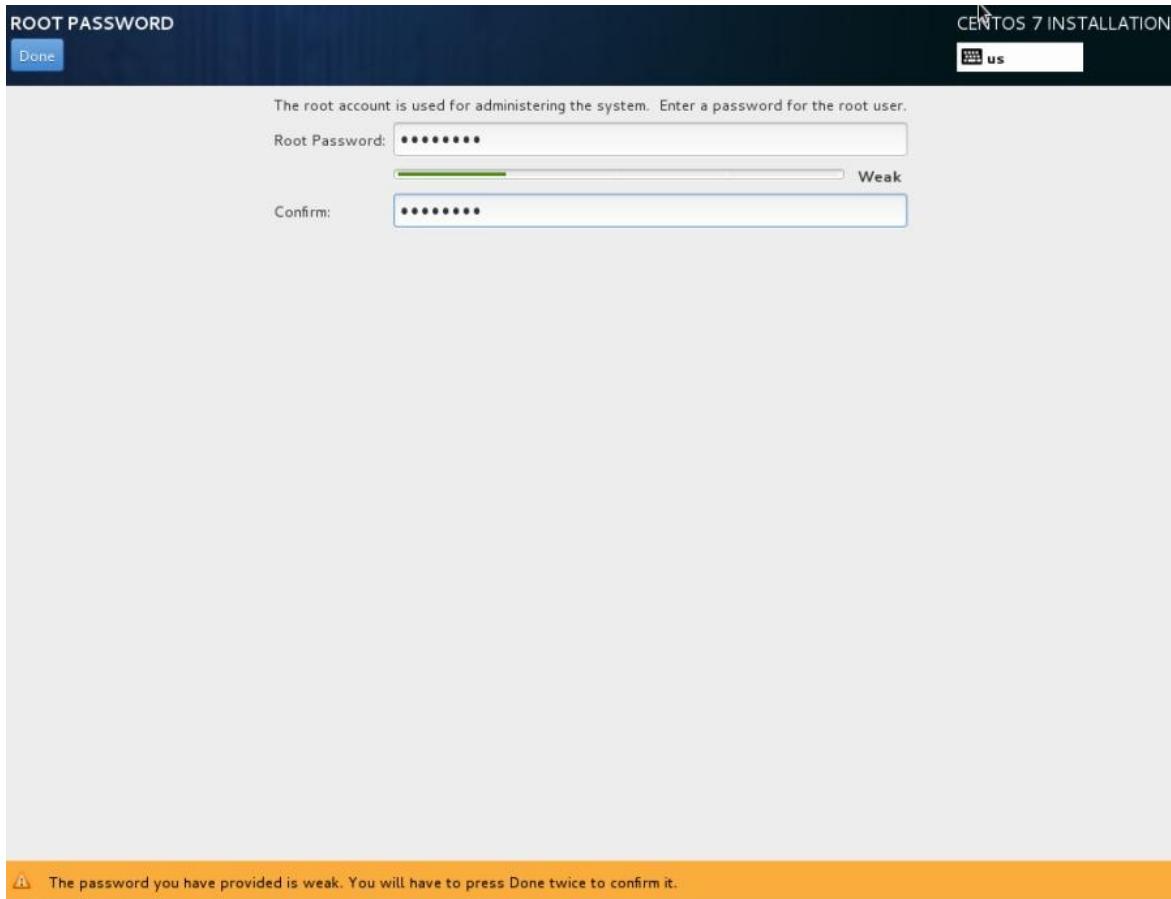
Step:8 Select the Software that you want to install. Click on "Software Selection". In my case i am selecting Gnome Desktop as shown below :



Step:9 Now Click on Begin Installation .



Step:10 Set root password



Step:11 Create a User

CREATE USER

CENTOS 7 INSTALLATION

us

Done

Full name LinuxTechi

Username linuxtechi

Tip: Keep your username shorter than 32 characters and do not use spaces.

Make this user administrator

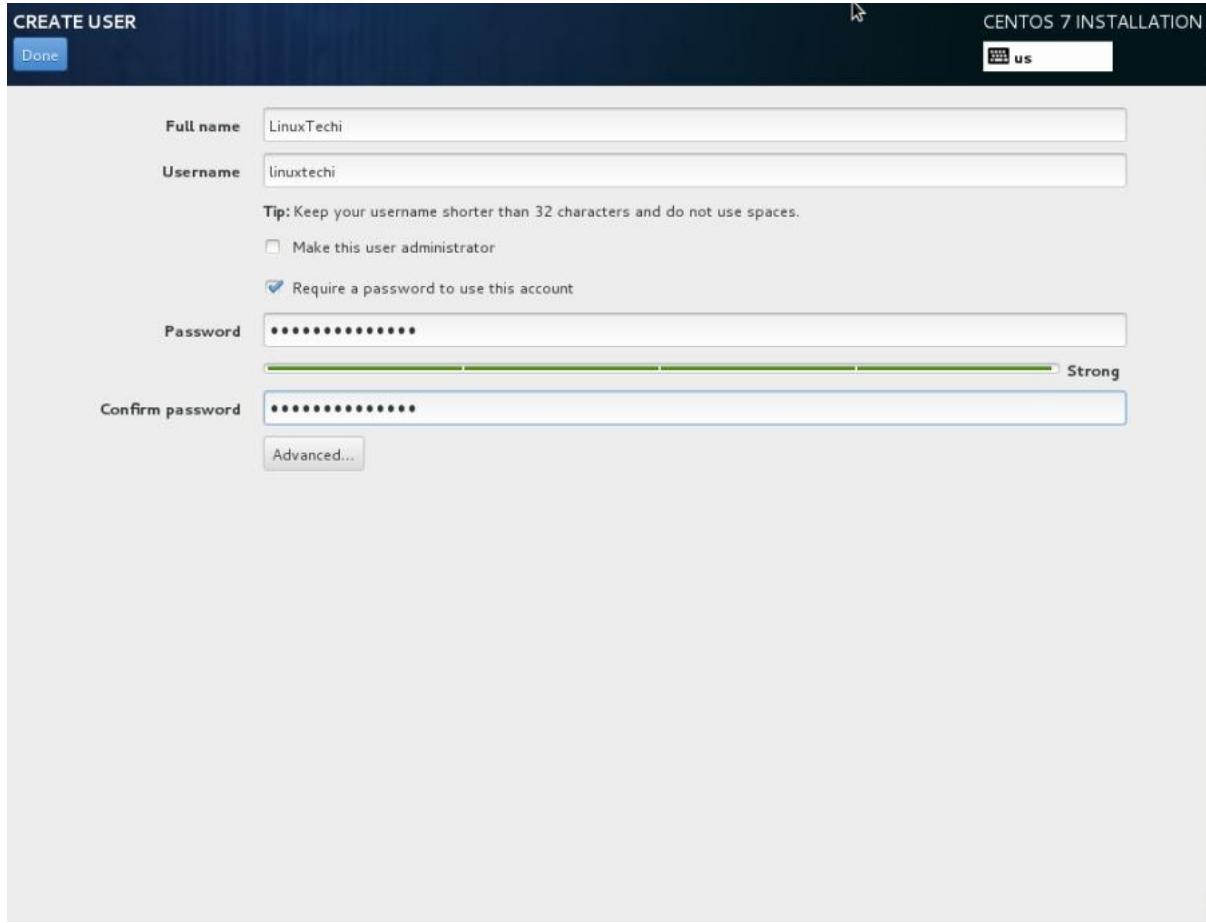
Require a password to use this account

Password *****

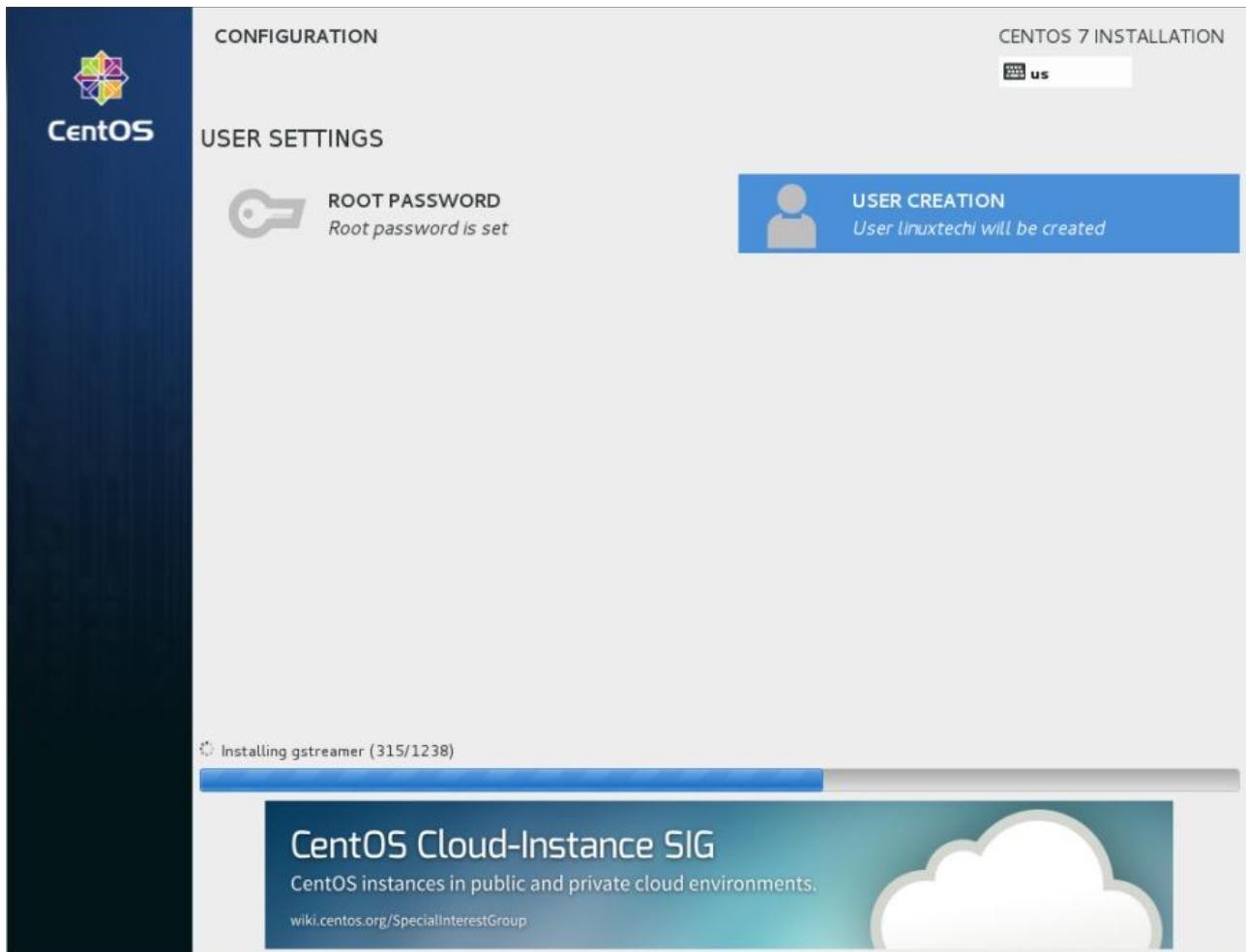
Confirm password *****

Strong

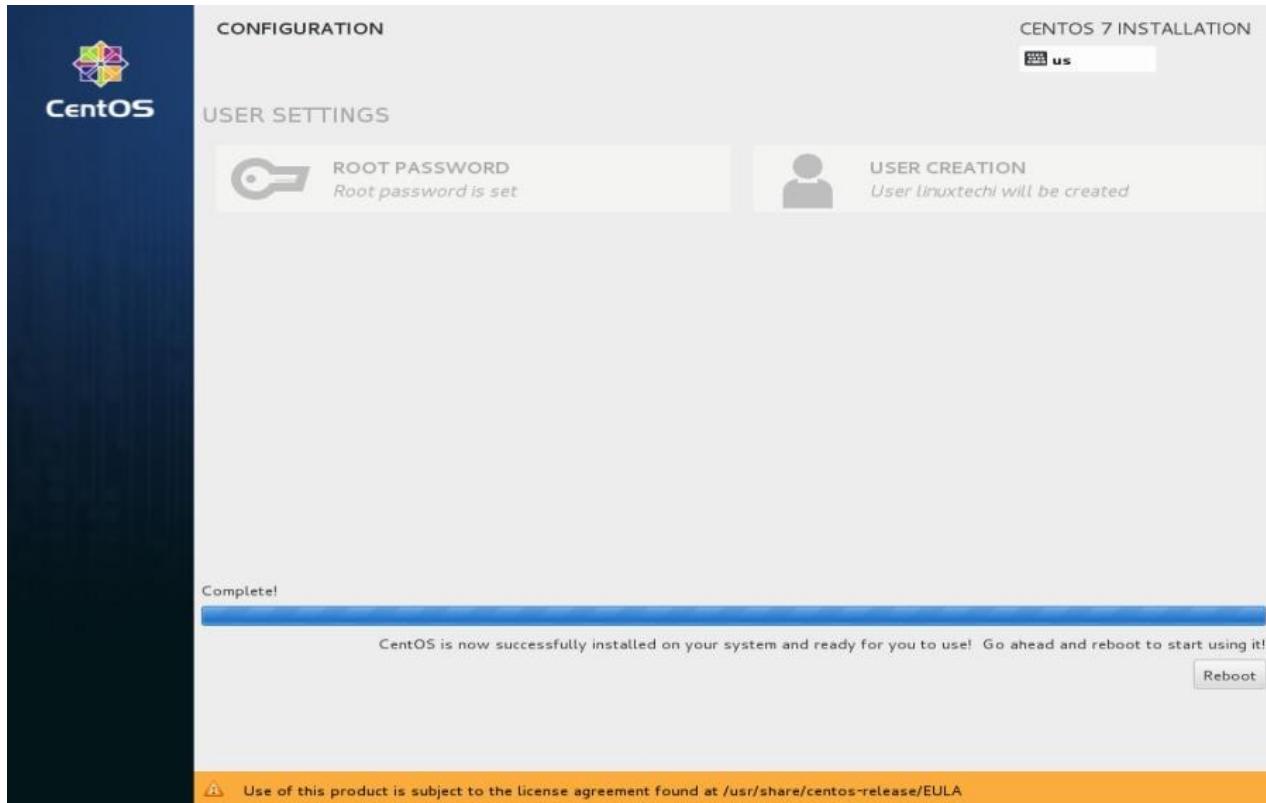
Advanced...



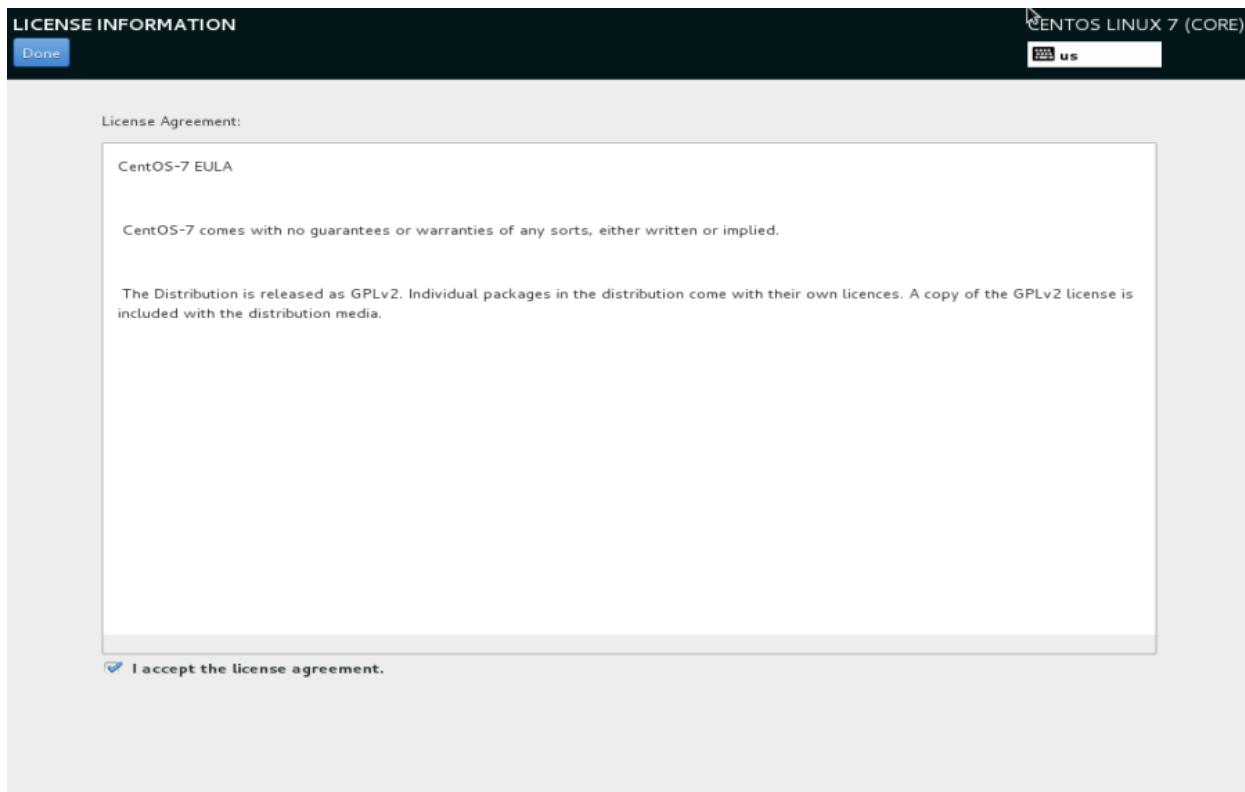
Step:12 Installation is in Progress as shown below



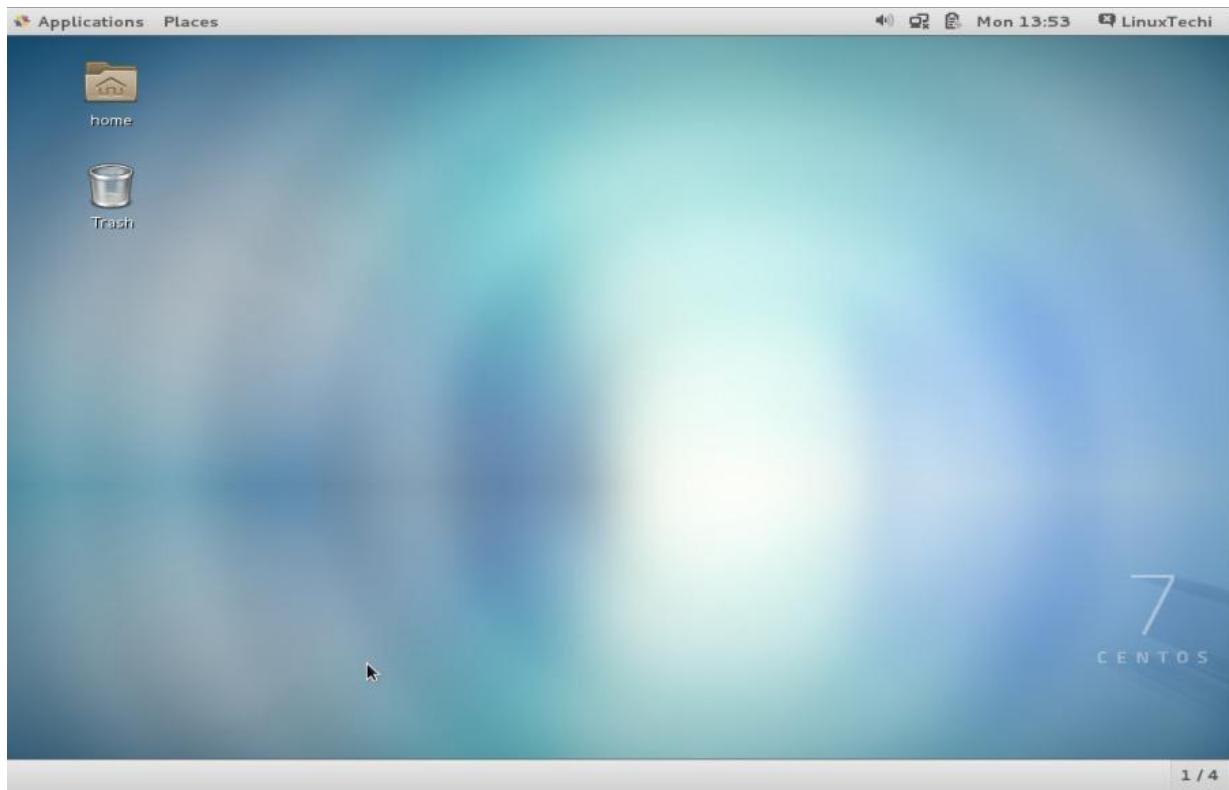
Once the installation is completed, you will be required to reboot the machine as shown below :



Step:13 When we first login to CentOS , Accept the EULA agreement



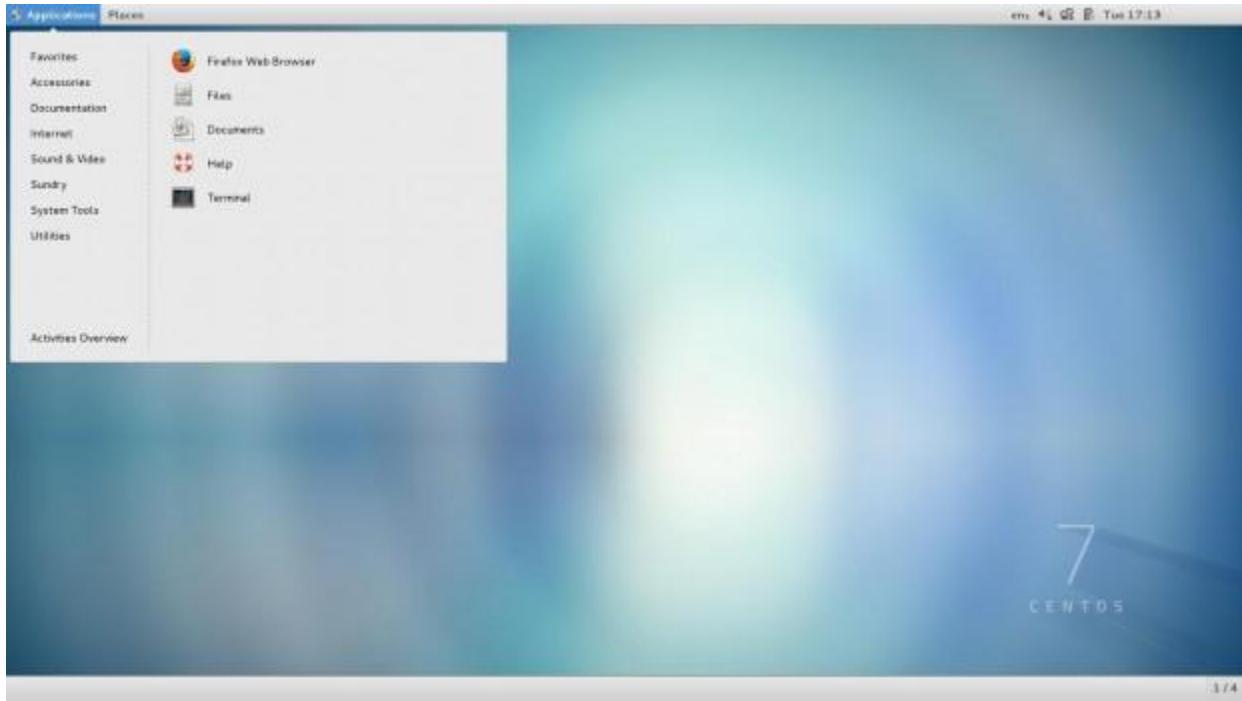
Below Screen will appear after login .



Another Example

Installation of “CentOS 7.0” with Screenshots

This tutorial will guide you on how to perform a minimal installation of latest version of **CentOS 7.0**, using the binary **DVD ISO** image, an installation that is best suitable for developing a future customizable server platform, with no Graphical User Interface, where you can install only the software that you need.



Installation of CentOS 7

If you want to find out more about what's new in this release of **CentOS 7.0** holds and download links, I suggest reading the previous article on release announcements:

1. [CentOS 7.0 Features and Download ISO Images](#)

Requirements

1. CentOS 7.0 DVD ISO

CentOS 7.0 Installation Process

1. After downloading the last version of CentOS using above links or using official [CentOS download](#) page. Burn it to a DVD or create a bootable USB stick using **LiveUSB Creator** called [Unetbootin](#).
2. After you have created the installer bootable media, place your DVD/USB into your system appropriate drive, start the computer, select your bootable unit and the first CentOS 7 prompt should appear. At the prompt choose **Install CentOS 7** and press **[Enter]** key.



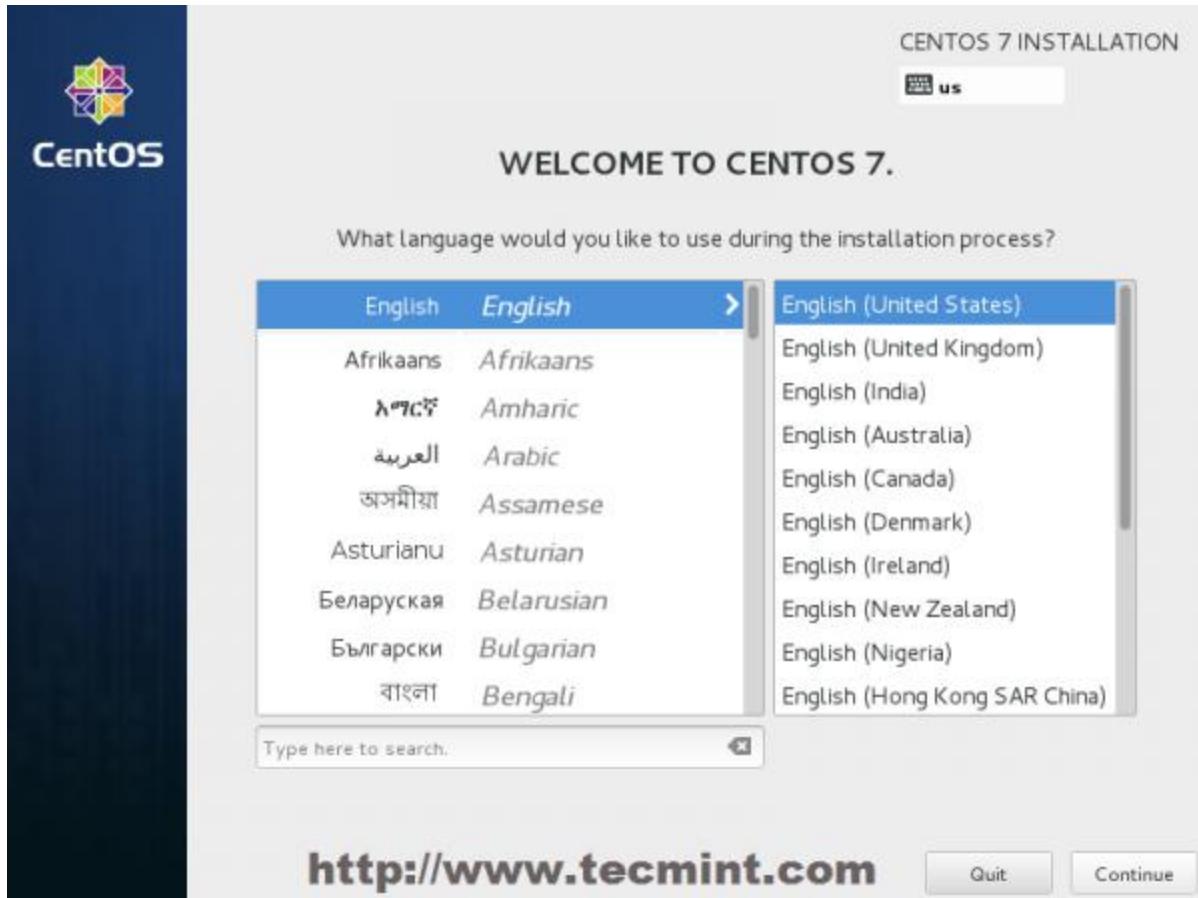
CentOS 7 Boot Menu

3. The system will start loading media installer and a Welcome screen should appear. Select your **Installation Process Language**, that will assist you through the entire installation procedure and click on **Continue**.

```
[ OK ] Started Configure read-only root support.
[ OK ] Started udev Coldplug all Devices.
      Starting udev Wait for Complete Device Initialization...
[ OK ] Started Import network configuration from initramfs.
[ OK ] Started Create static device nodes in /dev.
      Starting udev Kernel Device Manager...
[ OK ] Reached target Local File Systems (Pre).
[ OK ] Started udev Kernel Device Manager.
[ OK ] Started Device-Mapper Multipath Device Controller.
[ OK ] Started udev Wait for Complete Device Initialization.
      Starting Activation of DM RAID sets...
[ OK ] Started Activation of DM RAID sets.
[ OK ] Reached target Local File Systems.
      Starting Trigger Flushing of Journal to Persistent Storage...
      Starting Tell Plymouth To Write Out Runtime Data...
      Starting Create Volatile Files and Directories...
[ OK ] Reached target Encrypted Volumes.
[ OK ] Started Trigger Flushing of Journal to Persistent Storage.
[ OK ] Started Tell Plymouth To Write Out Runtime Data.
[ OK ] Started Create Volatile Files and Directories.
      Starting Update UTMP about System Reboot/Shutdown...
[ OK ] Started Update UTMP about System Reboot/Shutdown.
[ OK ] Reached target System Initialization.
[ OK ] Reached target Timers.
[ OK ] Listening on Open-iSCSI iscsid Socket.
[ OK ] Listening on Open-iSCSI iscsiuio Socket.
[ OK ] Listening on Avahi mDNS/DNS-SD Stack Activation Socket.
[ OK ] Listening on D-Bus System Message Bus Socket.
[ OK ] Reached target Sockets.
[ OK ] Reached target Basic System.
      Starting firewalld - dynamic firewall daemon...
      Starting Dump dmesg to /var/log/dmesg...
      Starting Terminate Plymouth Boot Screen...
      Starting System Logging Service...
      Starting Wait for Plymouth Boot Screen to Quit...
[ OK ] Started Dump dmesg to /var/log/dmesg.
```

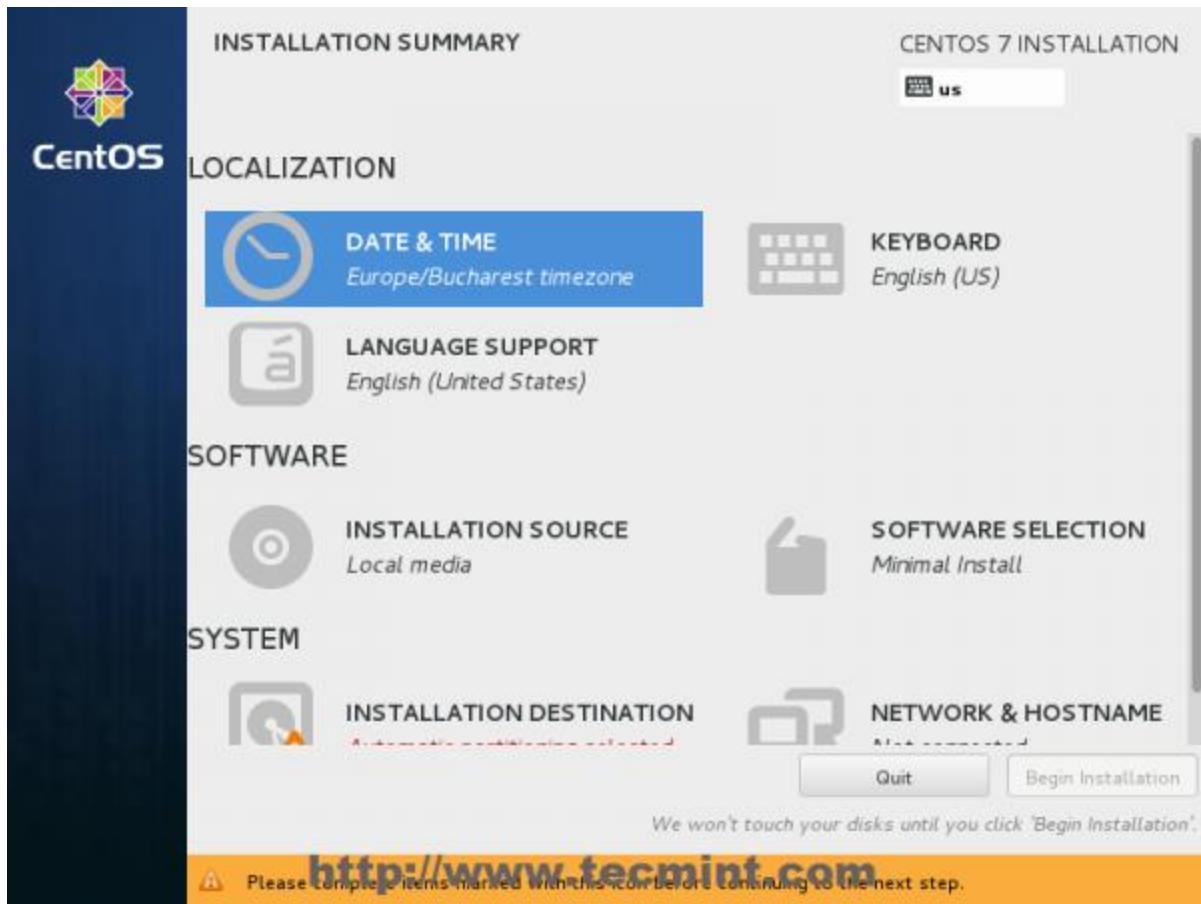
<http://www.tecmint.com>

CentOS Installer Loading



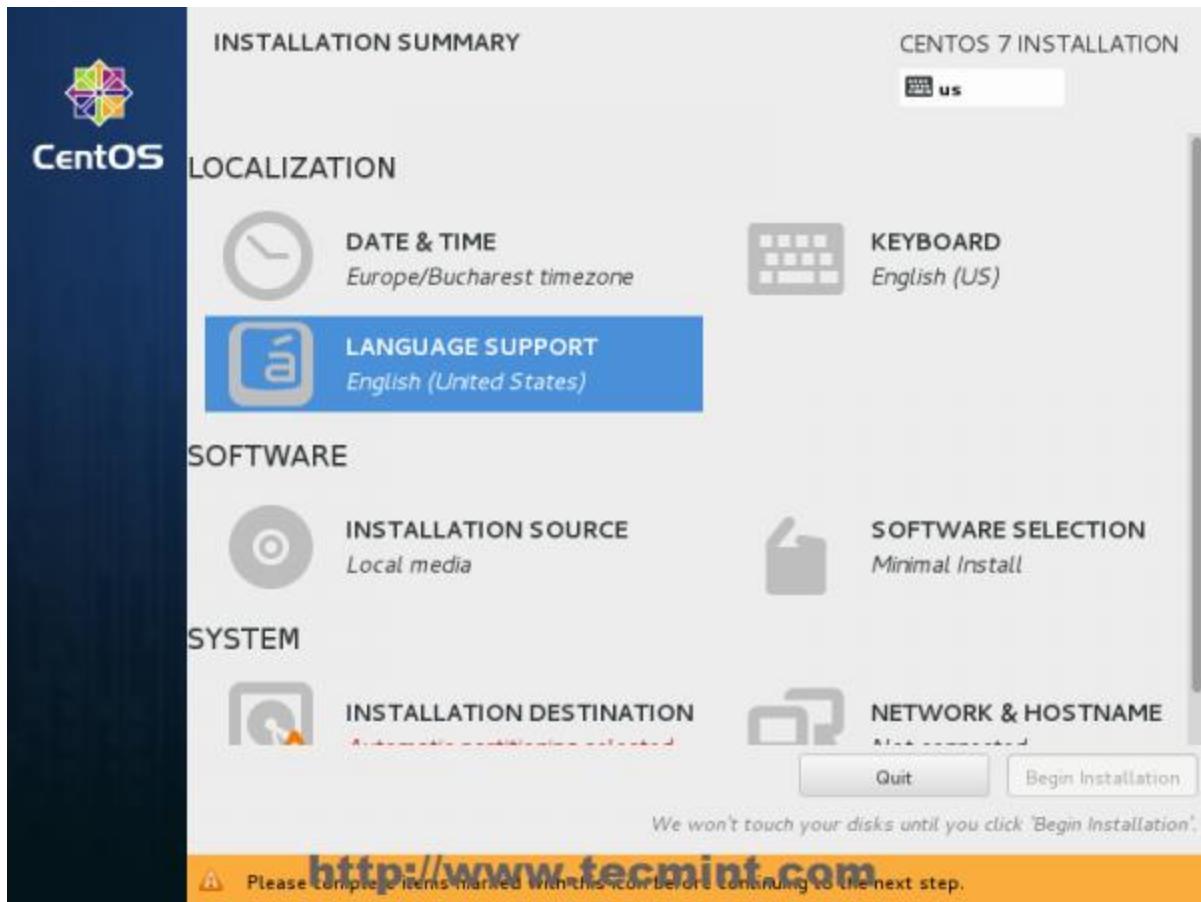
Select Installation Process Language

4. The next step, present screen prompt is **Installation Summary**. It contains a lot of options to fully customize your system. First thing you may want to setup is your time settings. Click on **Date & Time** and select your server physical location from the provided map and hit on upper **Done** button to apply configuration.



Select Date & Time and Location

5. The next step is to choose your **Language Support** and **Keyboard** settings. Choose your main and extra language for your system and when you're finished hit on **Done** button.



Select Language and Keyboard

Select additional language support to be installed:

Català

Catalan

Čeština Czech

Cymraeg Welsh

Dansk Danish

Deutsch German

Ελληνικά Greek

English English

Español Spanish

Eesti Estonian

Euskara Basque

فارسی Persian

Suomi Finnish

Espançol French

Type here to search: English (United States) English (United Kingdom) English (India) English (Australia) English (Canada) English (Denmark) English (Ireland) English (New Zealand) English (Nigeria) English (Hong Kong SAR China) English (Philippines) English (Singapore) English (South Africa)

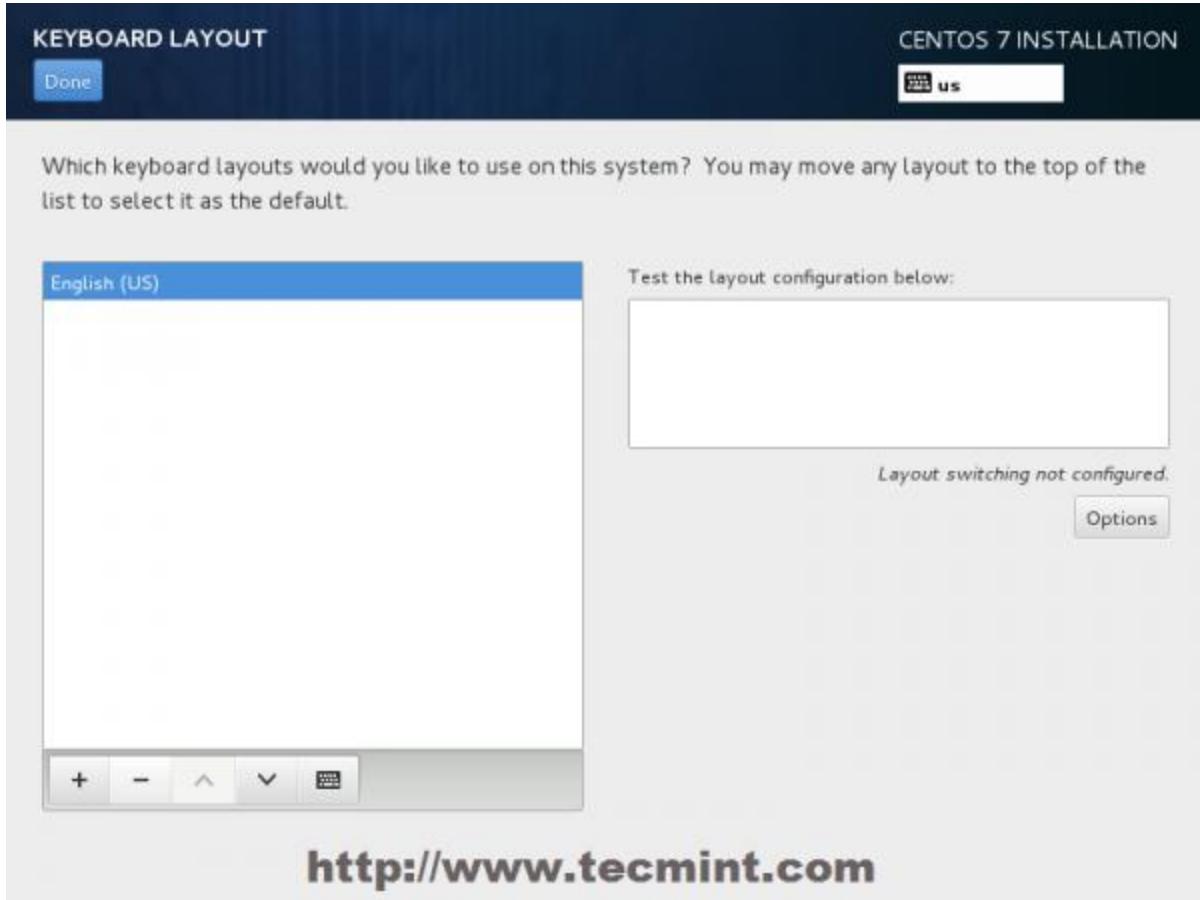
<http://www.tecmint.com>

Select English Language

6. The same way choose your **Keyboard Layout** by hitting the **plus** button and test your keyboard configuration using the right input filed. After you finish setting up your keyboard, again hit on upper **Done** button to apply changes and go back to main screen on Installation Summary.

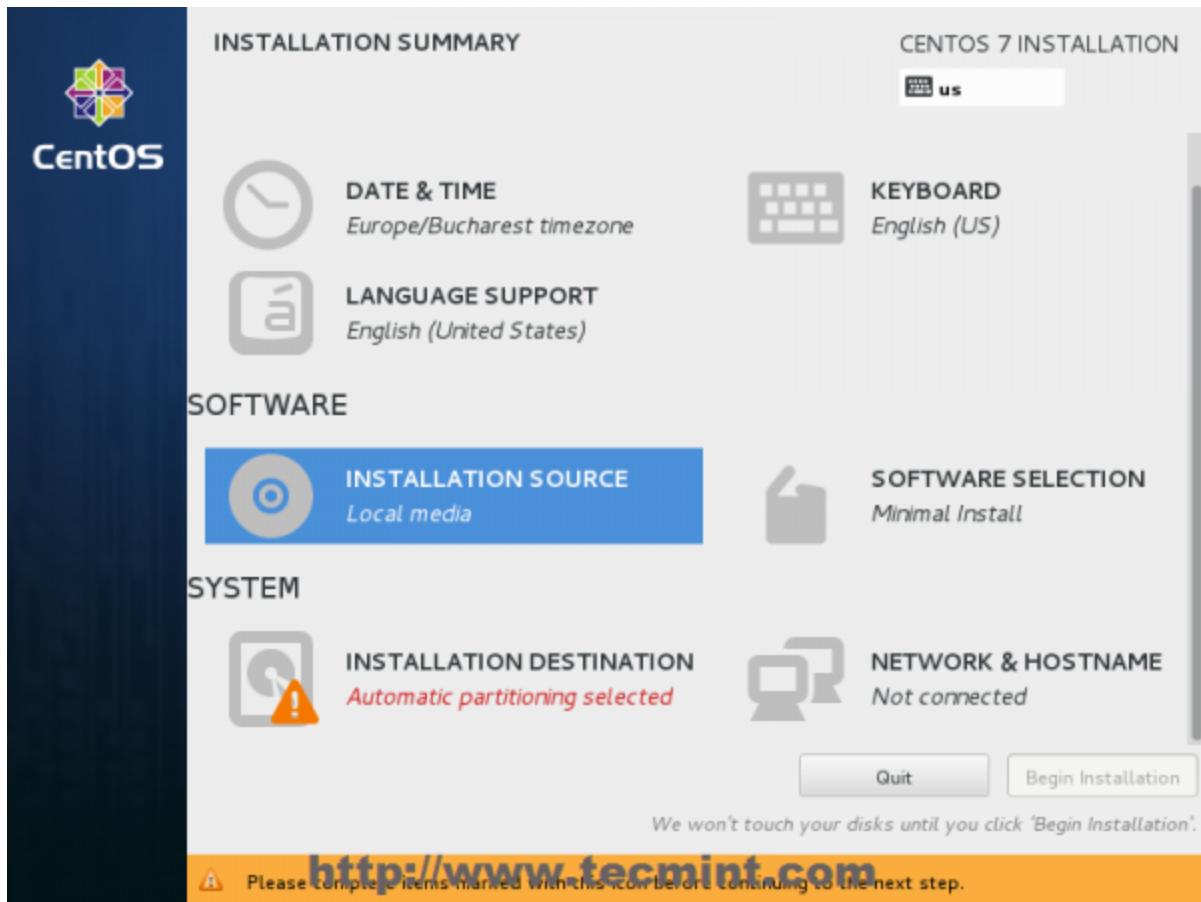


Choose Keyboard Layout

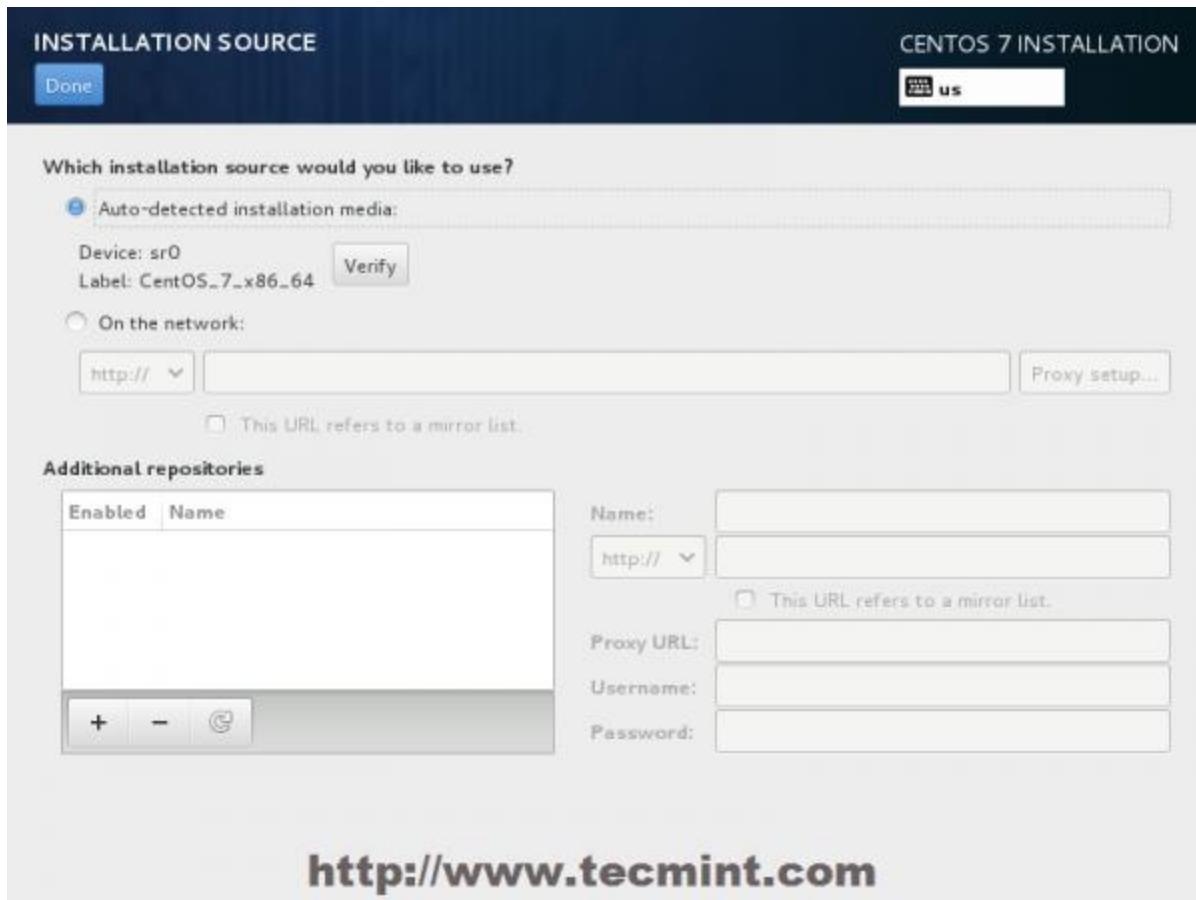


Choose English Keyboard

7. On the next step you can customize your installation by using other **Installation Sources** than your local DVD/USB media, such as a network locations using **HTTP**, **HTTPS**, **FTP** or **NFS** protocols and even add some additional repositories, but use this methods only if you know what you're doing. So leave the default **Auto-detected installation media** and hit on **Done** to continue.

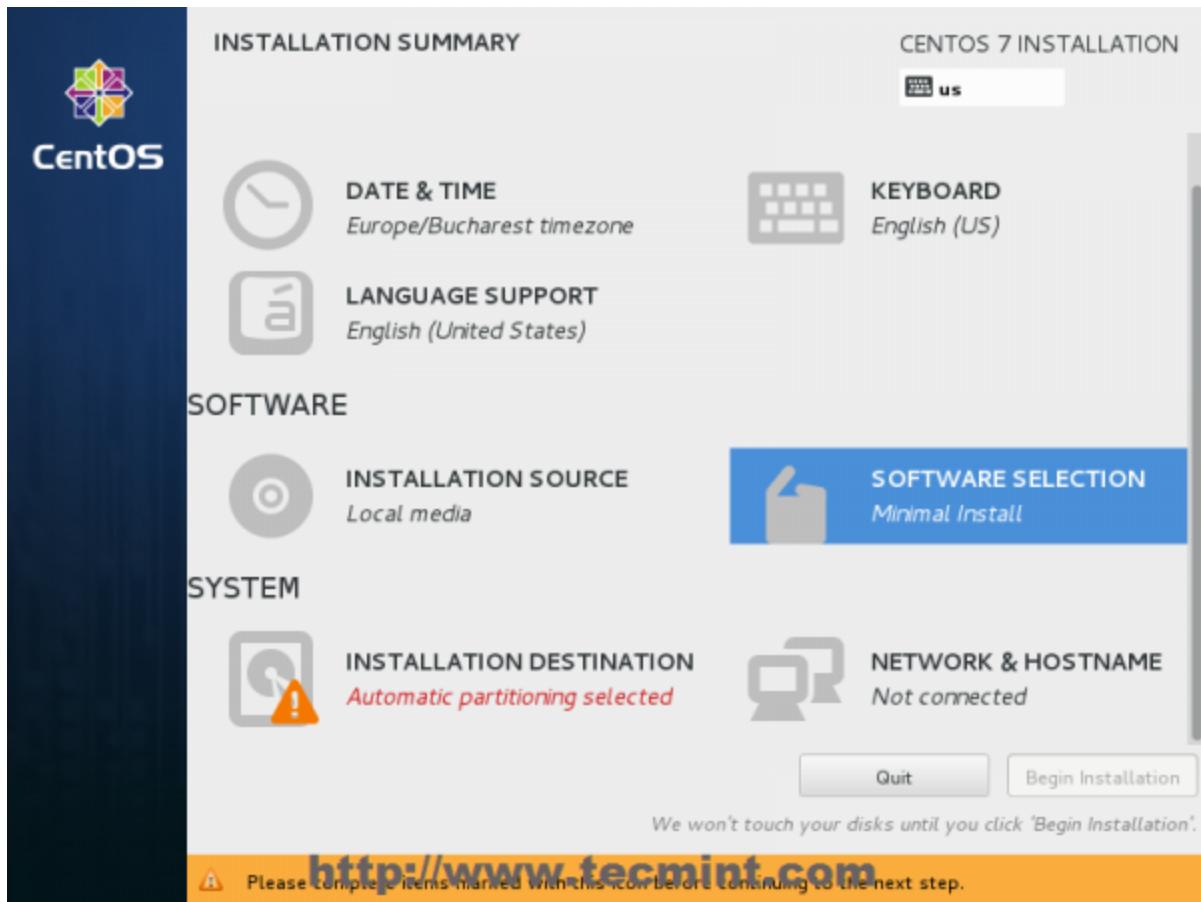


Choose Installation Sources



Auto Detect Installation Type

8. On the next step you can choose your system installation software. On this step CentOS offers a lot of Server and Desktop platform environments that you choose from, but, if you want a high degree of customization, especially if you are going to use CentOS 7 to run as a server platform, then I suggest you select **Minimal Install** with **Compatibility Libraries** as **Add-ons**, which will install a minimal basic system software and later you can add other packages as your needs require using **yum groupinstall** command.



Software Selection

SOFTWARE SELECTION

Done

CENTOS 7 INSTALLATION

FIPS US

Base Environment

Basic functionality.

 Infrastructure Server

Server for operating network infrastructure services.

 File and Print Server

File, print, and storage server for enterprises.

 Basic Web Server

Server for serving static and dynamic internet content.

 Virtualization Host

Minimal virtualization host.

 Server with GUI

Server for operating network infrastructure services, with a GUI.

 GNOME Desktop

GNOME is a highly intuitive and user friendly desktop environment.

 KDE Plasma Workspaces**Add-Ons for Selected Environment** **Compatibility Libraries**

Compatibility libraries for applications built on previous versions of CentOS Linux.

 Development Tools

A basic development environment.

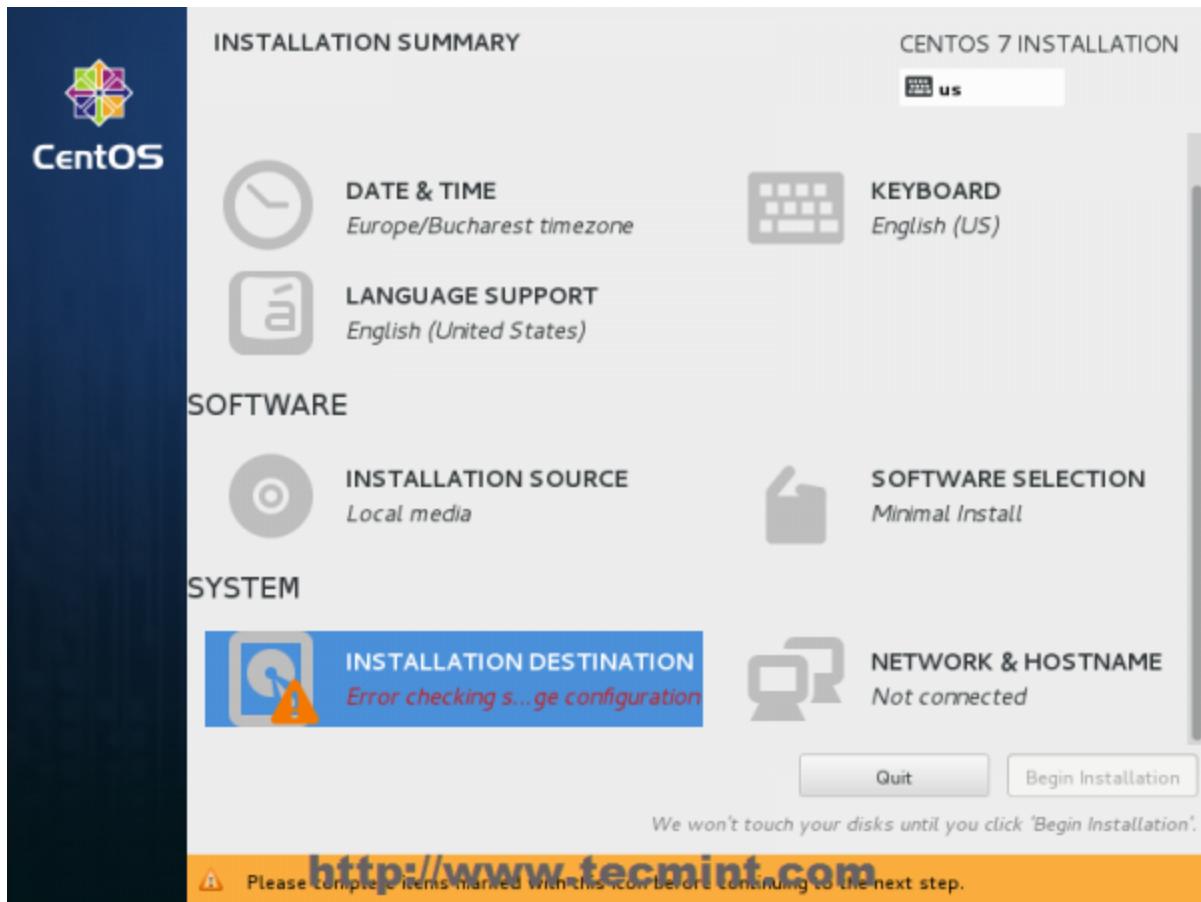
 Smart Card Support

Support for using smart card authentication.

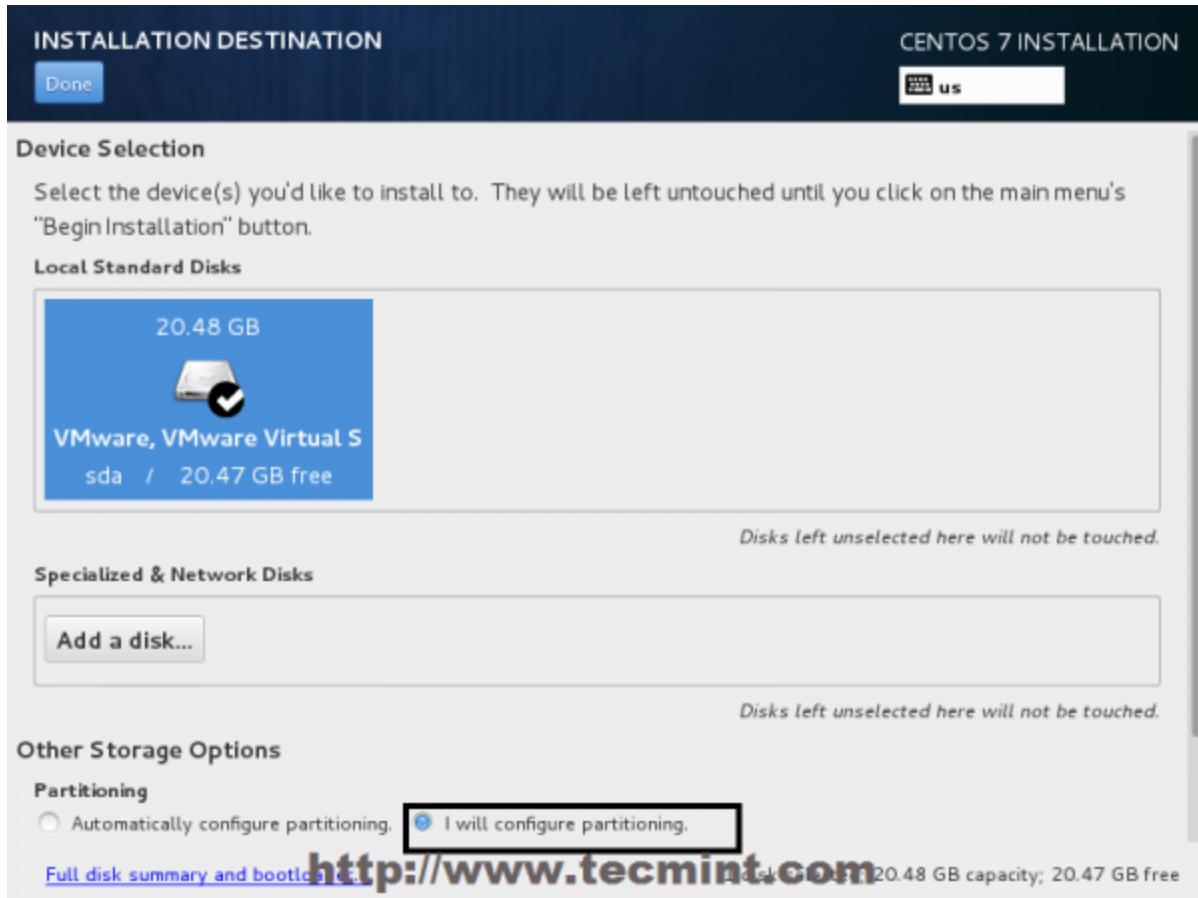
<http://www.tecmint.com>

Select CentOS 7 Minimal Install

9. Now it's time to partition your hard-drive. Click on **Installation Destination** menu, select your disk and choose **I will configure partitioning**.



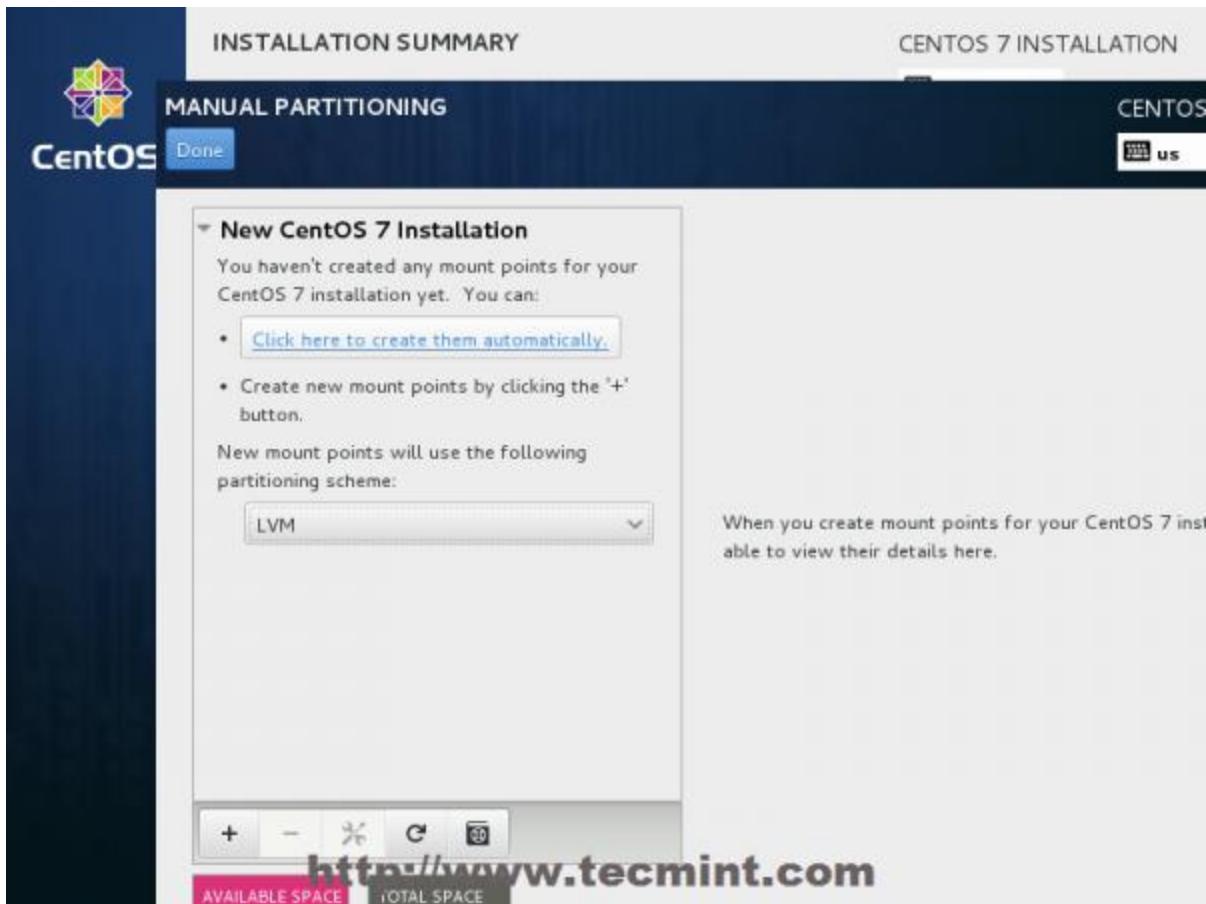
Choose Installation Destination



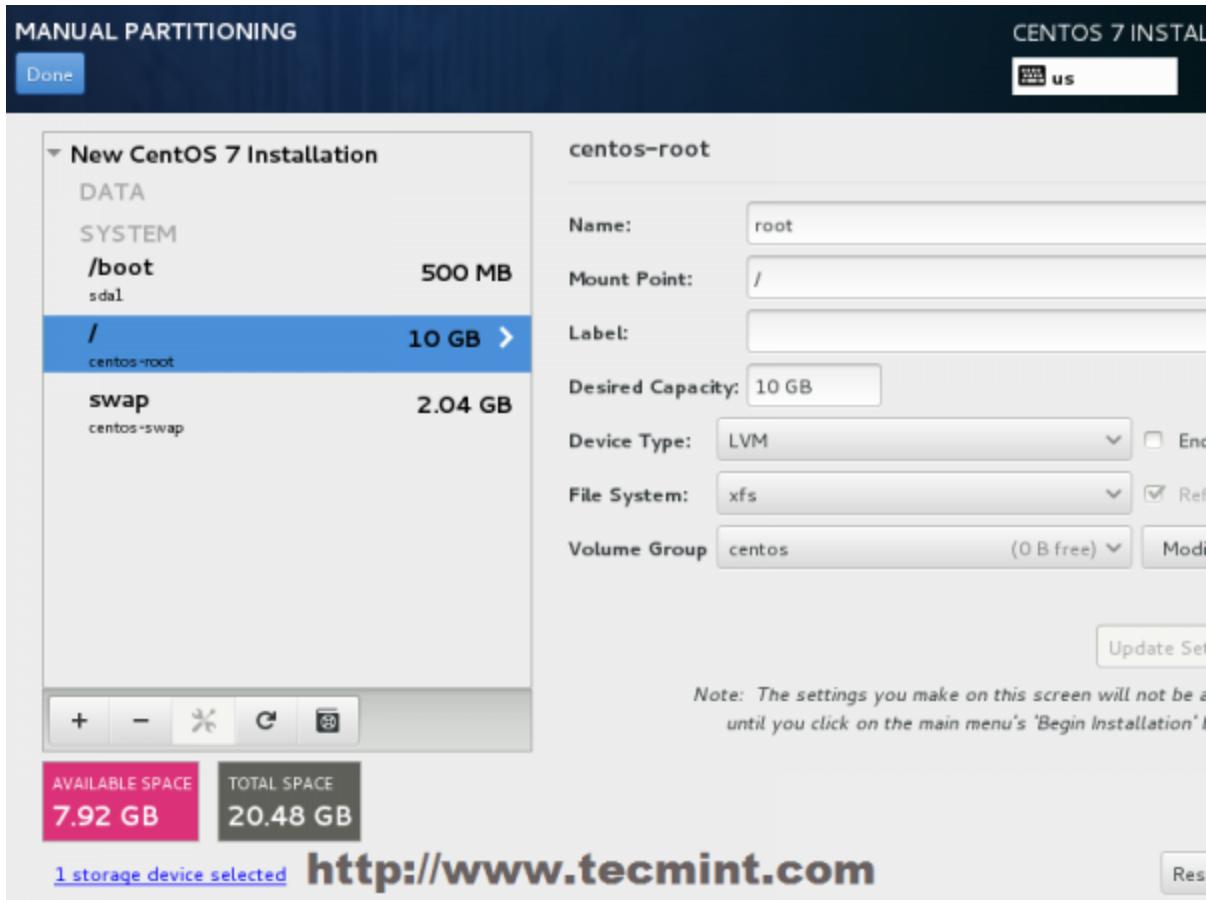
Installation Device Selection

10. On the next screen, choose **LVM** (Logical Volume Manager) as partition layout and, then, click on **Click here to create them automatically**, option which will create three system partition using **XFS** filesystem, automatically redistributing your hard-disk space and gathering all LVS into one big **Volume Group** named **centos**.

1. **/boot** – Non LVM
2. **/(root)** – LVM
3. **Swap** – LVM

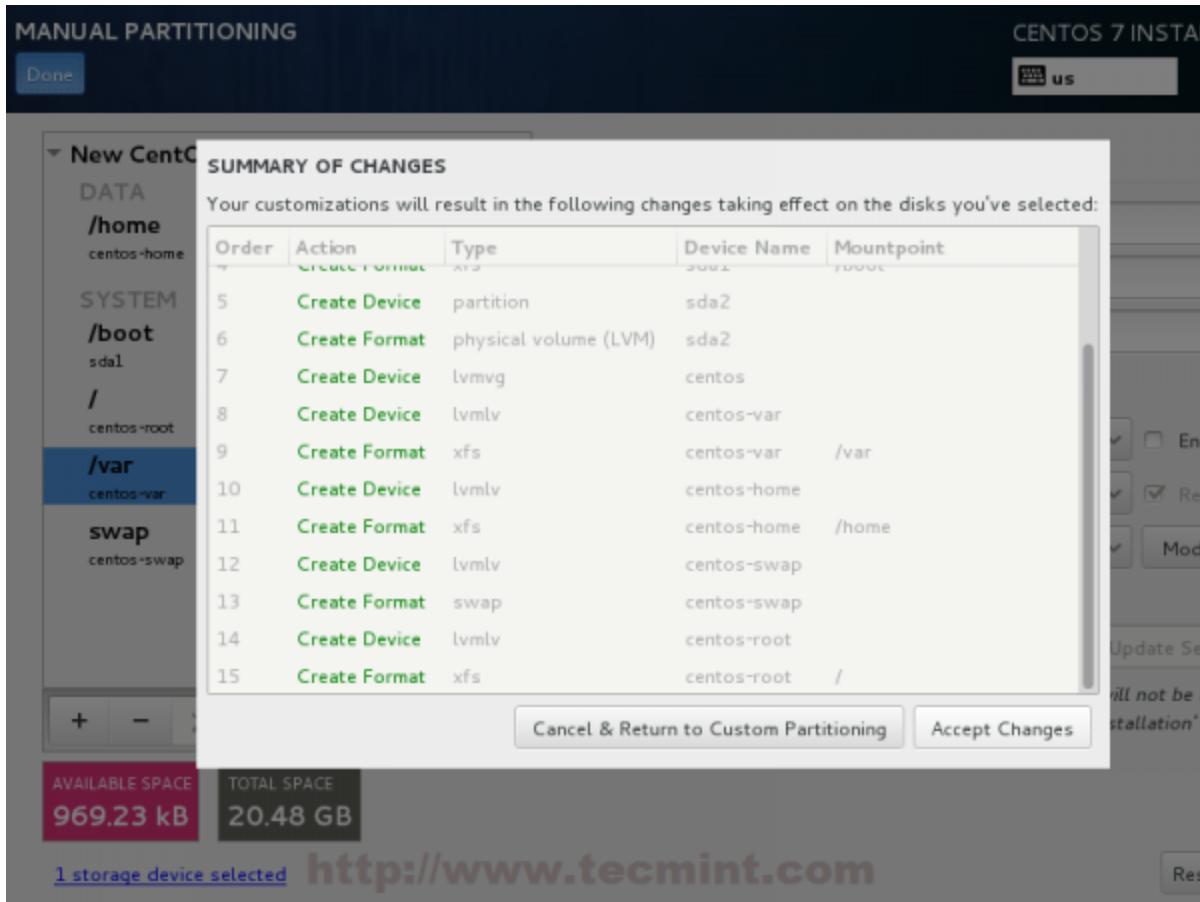


Select LVM Partition Type



Create Partitions

11. If you are not pleased with the default partition layout done automatically by the installer you can completely **add, modify or resize** your partition scheme and when you finish hit on **Done** button and **Accept Changes** on the Summary of Changes prompt.

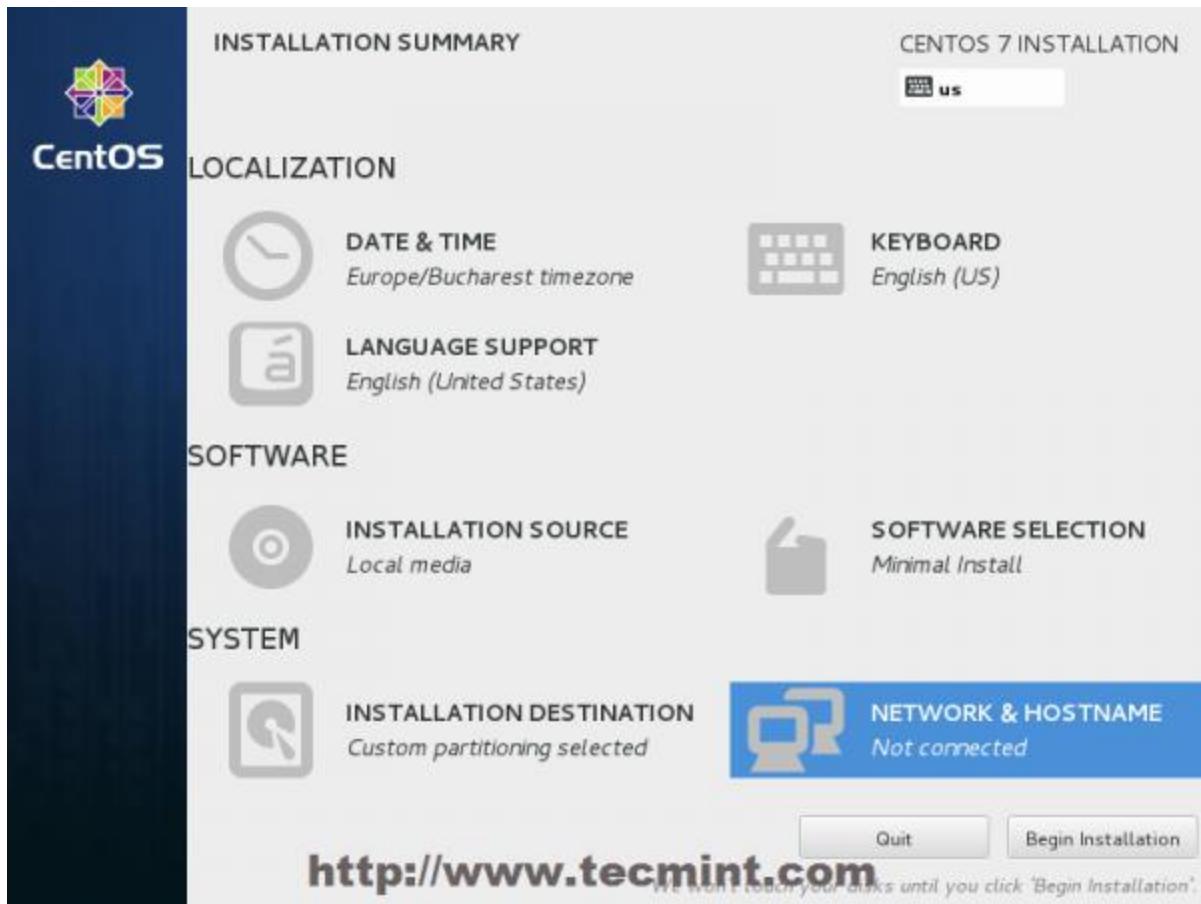


Summary of Partition Changes

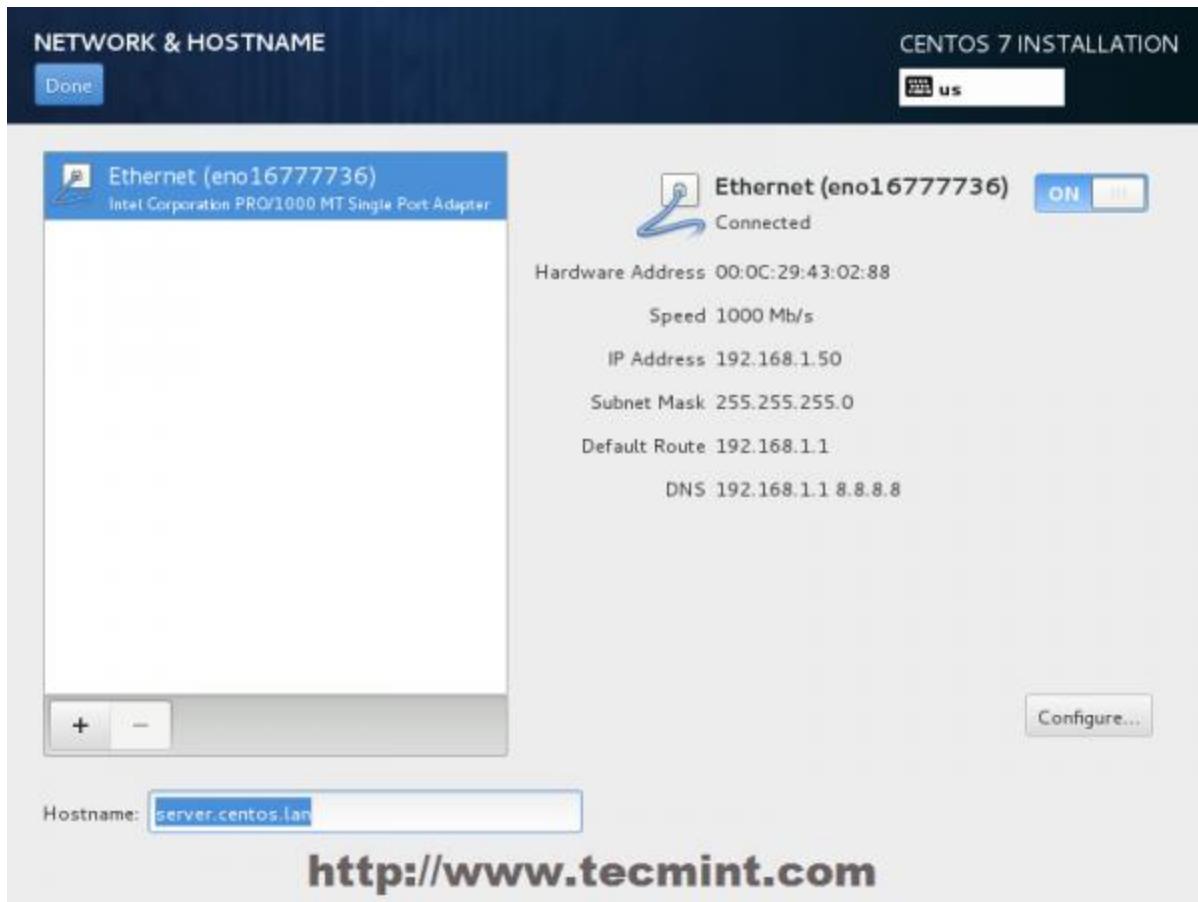
NOTE: For those users, who have hard-disks more than 2TB in size, the installer automatically will convert partition table to GPT, but if you wish to use GPT table on smaller disks than 2TB, then you should use the argument **inst.gpt** to the installer boot command line in order to change the default behaviour.

12. The next step is to set your system hostname and enable networking. Click on **Network & Hostname** label and type your system **FQDN** (Fully Qualified Domain Name) on Hostname filed, then enable your Network interface, switching the top **Ethernet** button to **ON**.

If you have a functional DHCP server on your network then it will automatically configure all your network setting for enabled NIC, which should appear under your active interface.

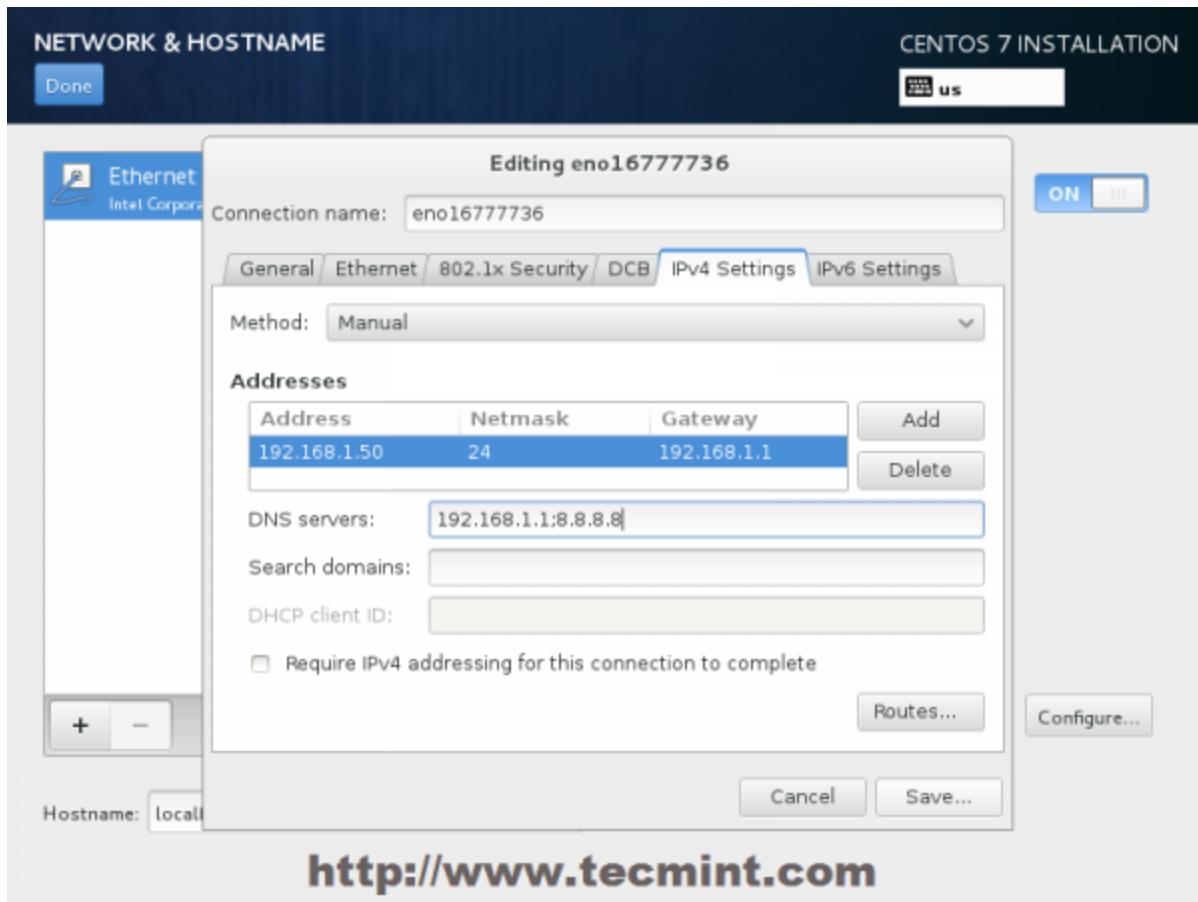


Set System Hostname



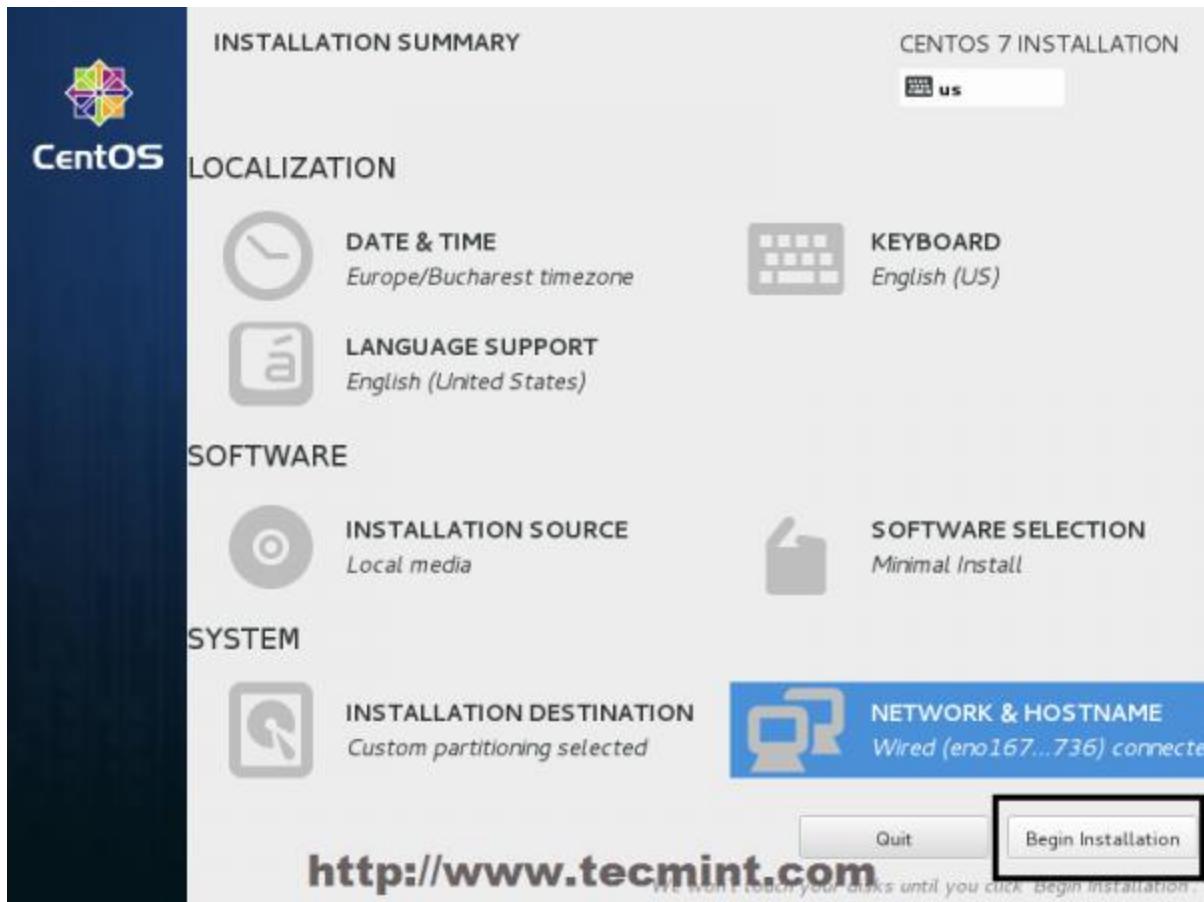
Enable Ethernet Interface

13. If your system will be destined as a server it's better to set static network configuration on Ethernet NIC by clicking on **Configure** button and add all your static interface settings like in the screenshot below, and when you're finished hit on **Save** button, disable and enable Ethernet card by switching the button to **OFF** and **ON**, and, then hit on **Done** to apply setting and go back to main menu.

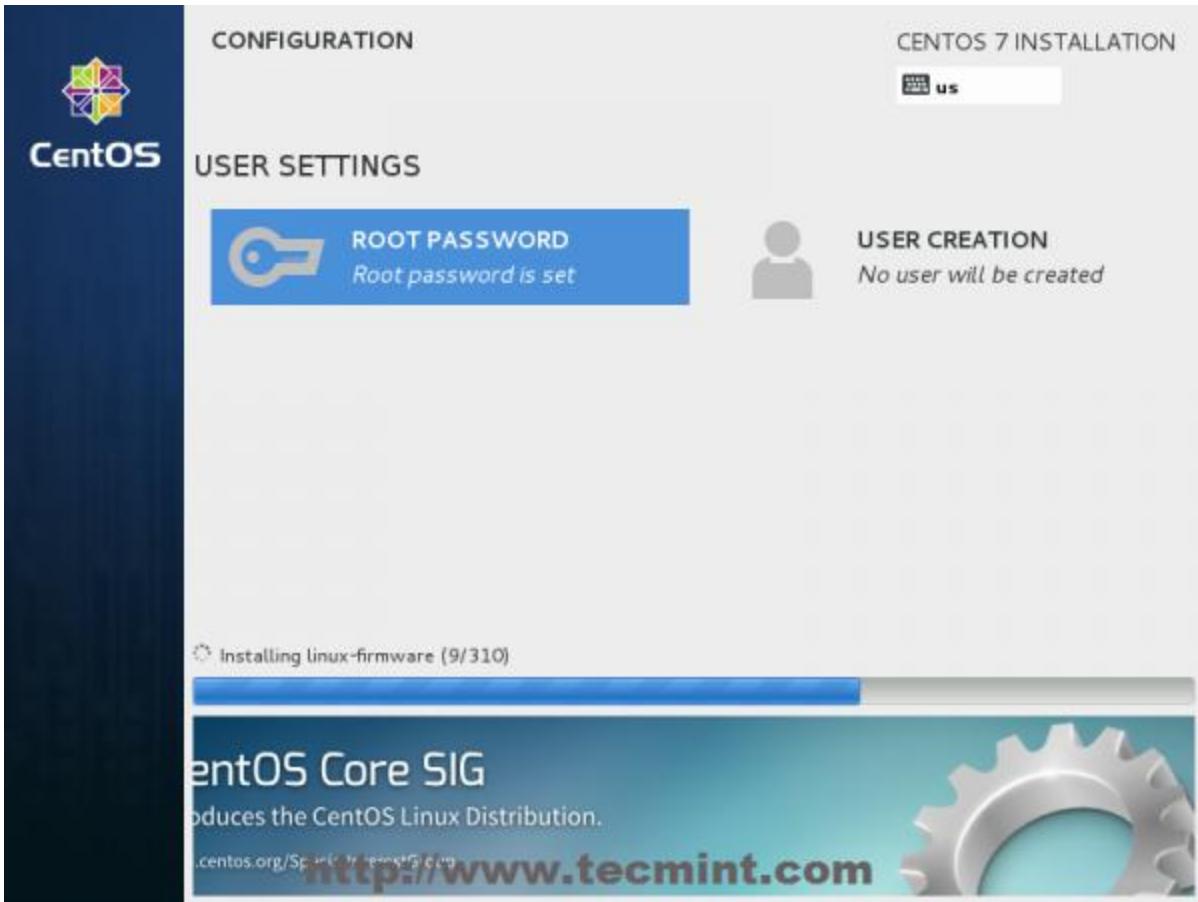


Enter Network Settings

- 14.** Now it's time to start installation process by pressing on **Begin Installation** button and set up a strong password for **root** account.



Click on Begin Installation

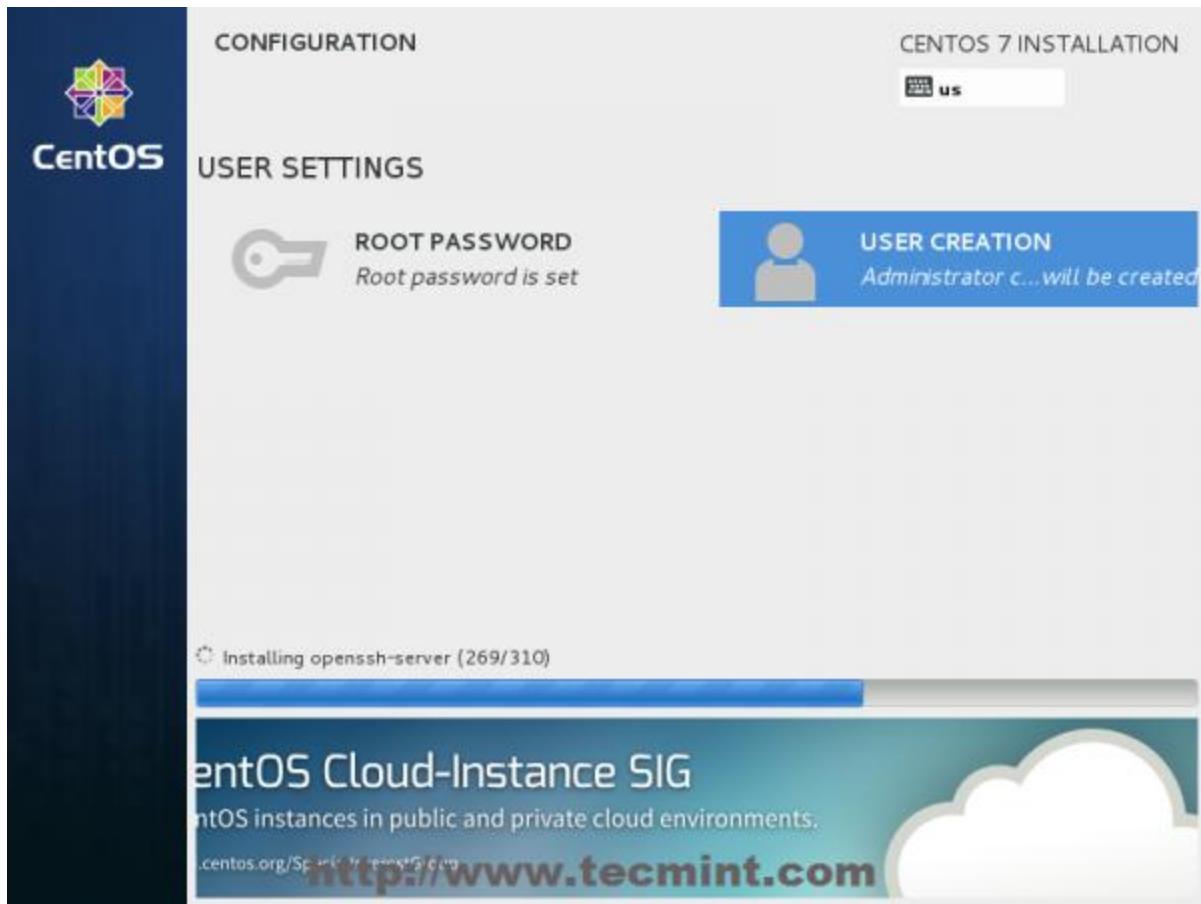


Select Root Password

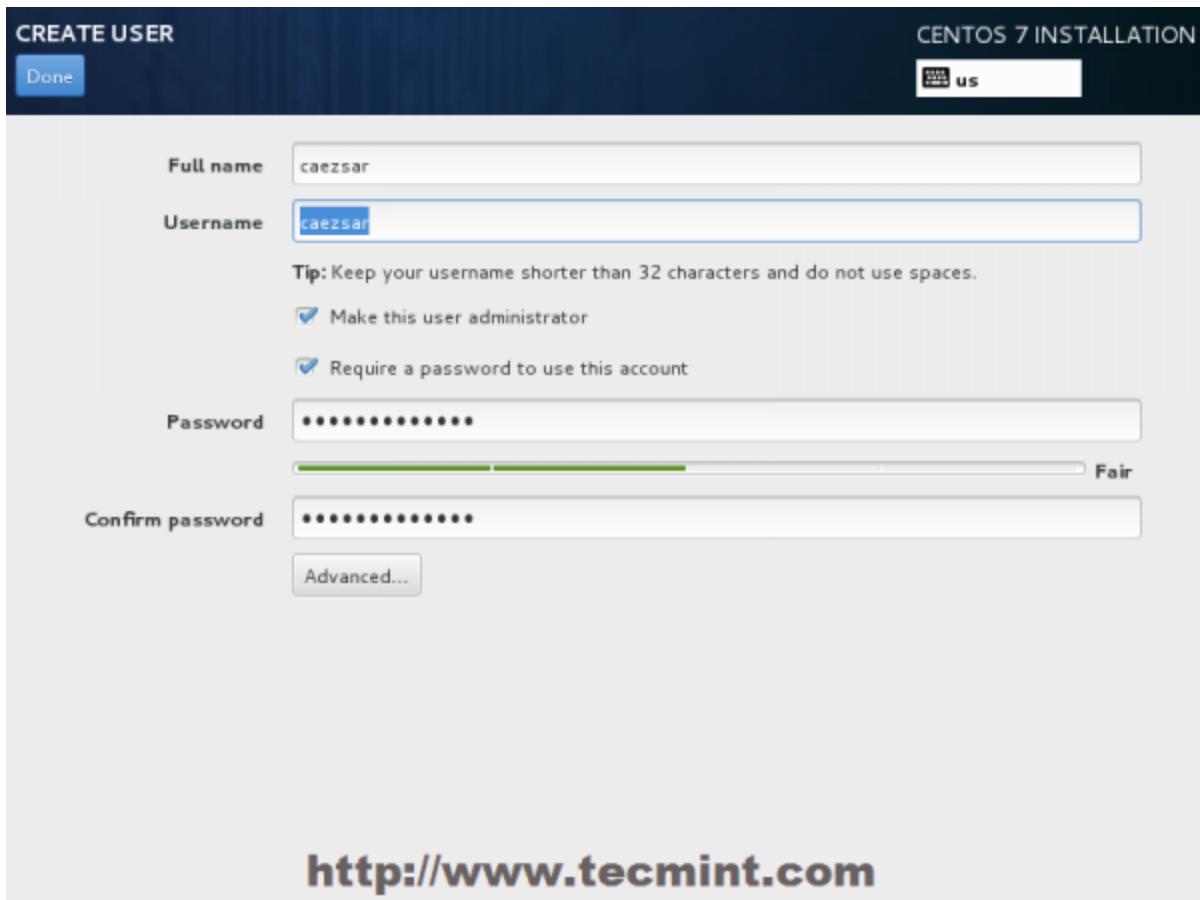


Enter Root Password

15. After you finish setting up a strong password for root account move to **User Creation** and create your first system user. You can designate this user to become a System Admin with root privileges using **sudo** command by checking the box **Make this user administrator**, then click on **Done** to go back on main menu and wait for the installation process to finish.

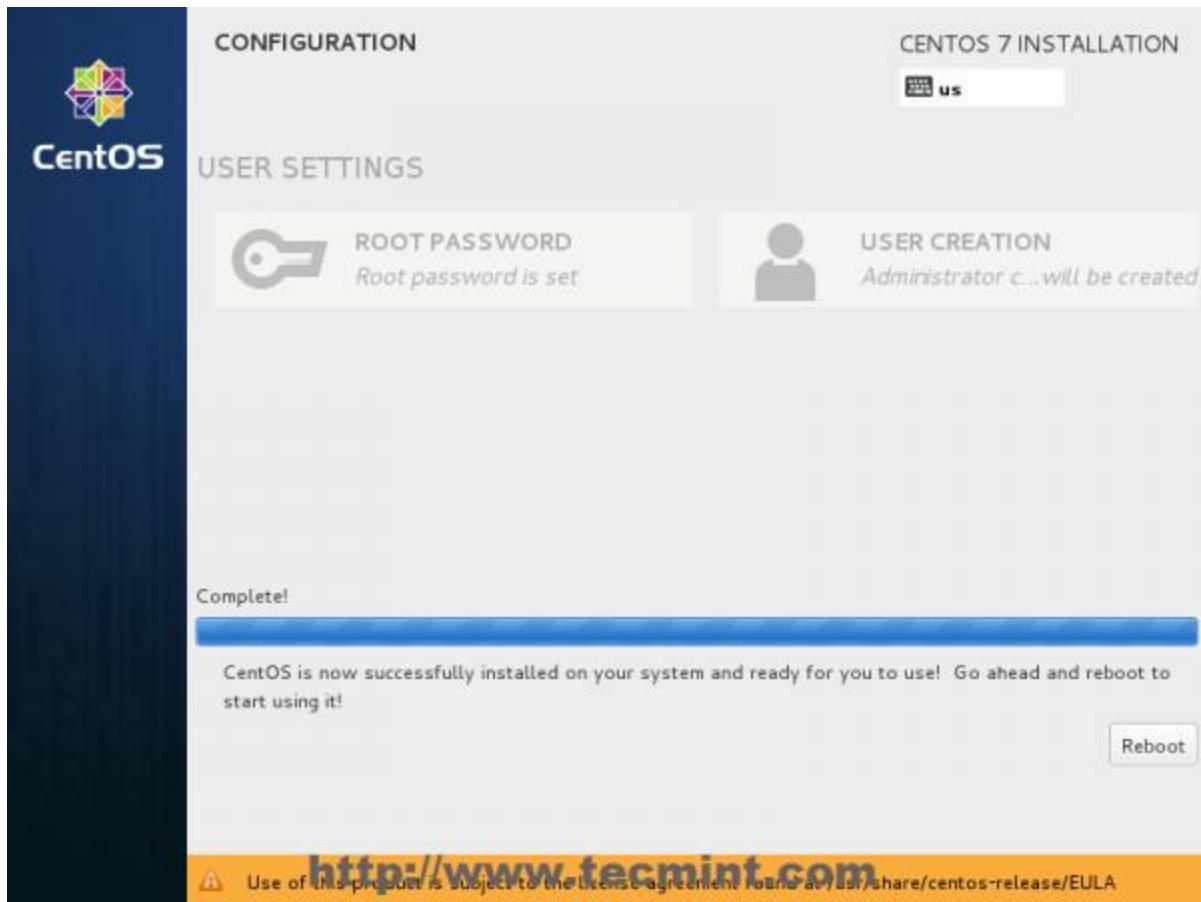


CentOS 7 Installation Process



User Creation and Set Password

- 16.** After the installation process finishes, the installer will show a successfully message on screen, demanding to reboot your system in order to use it.



CentOS 7 Installation Complete

Congratulation! You have now installed last version of **CentOS** on your bare new machine. Remove any installation media and **reboot** your computer so you can login to your new minimal **CentOS 7** environment and perform other system tasks, such as update you system and install other useful software needed to run day to day tasks.

Disable Firewall:

Switch user to root (# su root) put password

[How to Stop and Disable Firewalld on CentOS 7](#)

It is **highly** recommended that you have another firewall protecting your network or server before, or immediately after, disabling firewalld.

Pre-Flight Check

- These instructions are intended specifically for stopping and disabling firewalld CentOS 7.
- I'll be working from a Liquid Web Self Managed CentOS 7 server, and I'll be logged in as root.

Disable Firewalld

To disable firewalld, run the following command as root:

```
systemctl disable firewalld
```

Stop Firewalld

To stop firewalld, run the following command as root:

```
systemctl stop firewalld
```

Check the Status of Firewalld

To check the status of firewalld, run the following command as root:

```
systemctl status firewalld
```

Wait, you actually wanted to Start and Enable Firewalld on CentOS 7? Then hit our tutorial on:
[How to Start and Enable Firewalld on CentOS 7!](#)

How to Setup network on centos 7

Setup network on centos 7

let's start, Type “**nmcli d**” command in your terminal for quick identification of Ethernet cards installed in your machine.

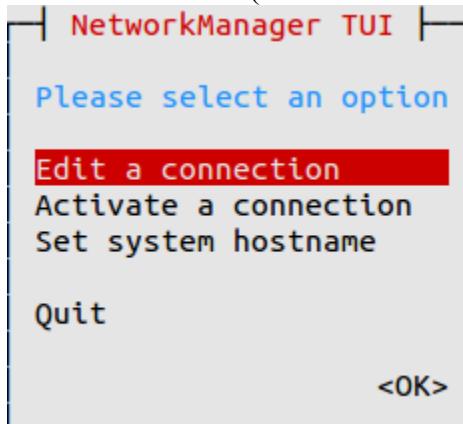
DEVICE	TYPE	STATE	CONNECTION
enp0s17	ethernet	disconnected	--
enp0s18	ethernet	disconnected	--
lo	loopback	unmanaged	--

Here we have 2 interfaces named “**enp0s17**” and “**enp0s18**” . it might be different in your case (Eg: **em1** or **p4p1**).

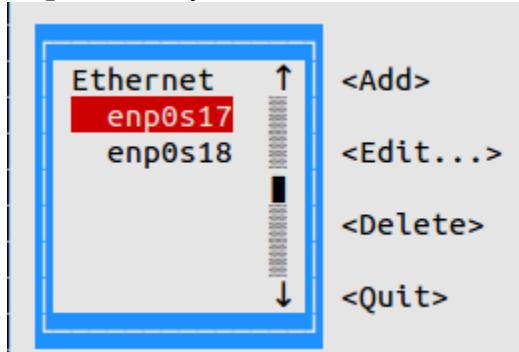
GUI Mode

Recommended for beginners

Step 1 » Type this command “**nmtui**” to open Network manager and press enter after choosing ”Edit a connection” (Use TAB for choosing options) .



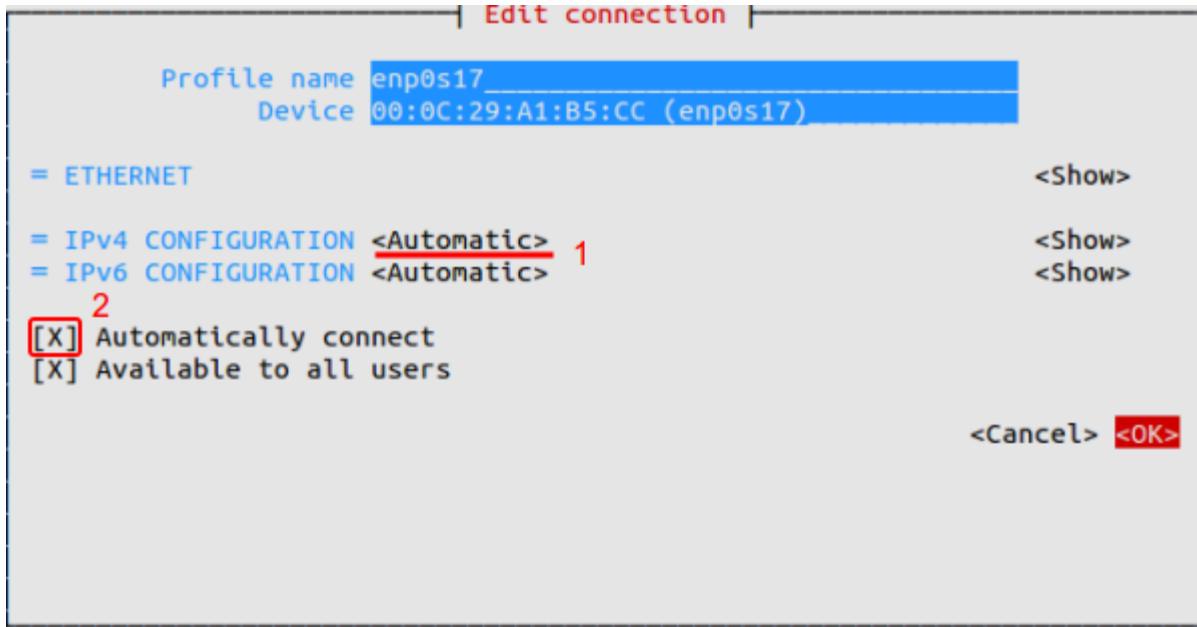
Step 2 » Now you can see all network interfaces, choose one and click “**Edit**“.



» *DHCP configuration*

Step 3 » For DHCP,

1. Choose “**Automatic**” in IPv4 CONFIGURATION.
2. Choose Automatic Connect check box.
3. Press OK and quit Network manager.



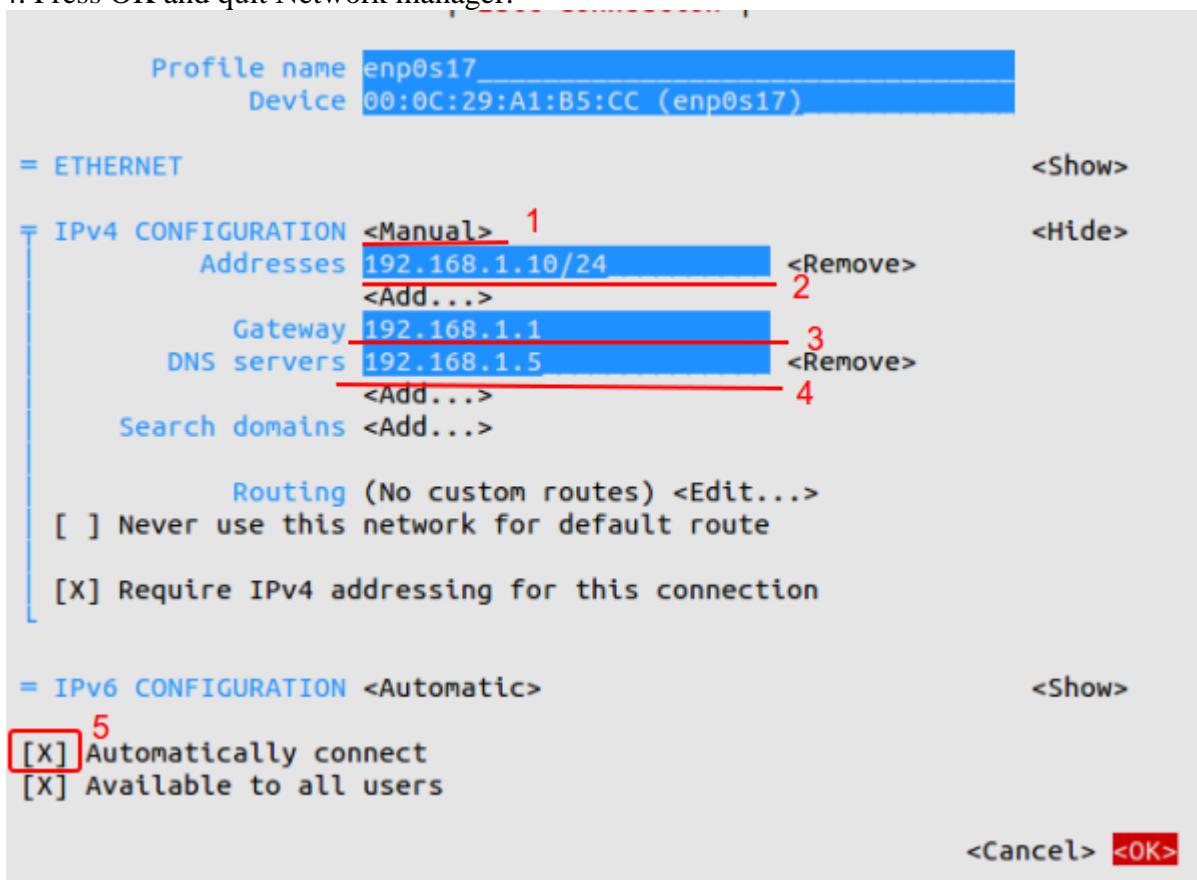
Now Restart network service by typing below command.

`systemctl restart network` Now your server will get IP Address from DHCP .

» *Static configuration*

Step 4 » For manual IP address,

1. Choose “**Manual**” in IPv4 CONFIGURATION.
2. Add IP Address with Subnet , Gateway and DNS server (Refer below image).
3. Choose Automatic Connect check box.
4. Press OK and quit Network manager.



Now Restart network service by typing below command.

systemctl restart network That's it, Interface will have static IP.

Command Mode

Step 1 » Network interface config files are located in **/etc/sysconfig/network-scripts/** directory.

Open ifcfg-enp0s17 file (For interface enp0s17) and you can see the content like below.

```
[root@krizna ~]# vi or nano /etc/sysconfig/network-scripts/ifcfg-enp0s17
TYPE=Ethernet
BOOTPROTO=none
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
NAME=enp0s17
UUID=7f1aff2d-b154-4436-9497-e3a4dedddcef
ONBOOT=no
HWADDR=00:0C:29:A1:B5:D6
PEERDNS=yes
PEERROUTES=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
```

» *DHCP configuration*

Step 2 » For DHCP

Find the below lines in config File.

```
BOOTPROTO=none
ONBOOT=no and replace with
BOOTPROTO=dhcp
```

ONBOOT=yes Now Restart network service by typing below command.

systemctl restart network Now your server will get IP Address from DHCP

» *Static configuration*

Step 3 » For Static IP.

Find the below lines in config File.

```
BOOTPROTO=none
ONBOOT=no and replace with
BOOTPROTO=static
ONBOOT=yes And add the below lines at the end of the file.
```

```
IPADDR=172.27.0.32
NETMASK=255.255.255.0
GATEWAY=172.27.0.1
```

DNS1=172.27.0.5 File will look like below after changes.

```
TYPE=Ethernet
BOOTPROTO=static
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
NAME=enp0s17
UUID=f0c5b37d-299a-43cb-b74b-618bb252d129
```

```
ONBOOT=yes
HWADDR=00:0C:29:A1:B5:CC
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPADDR=192.168.1.10
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
DNS1=192.168.1.5
```

Now Restart network service by typing below command.

```
systemctl restart network
```

Now Interface will have static IP.

Additionally you can use **/etc/sysconfig/network** file for hostname and DNS .

```
HOSTNAME=server.krizna.com
DNS1=192.168.1.5
DNS2=8.8.8.8
SEARCH=krizna.com
```

Have a nice day

SSH Server and Client Setup

Configure SSH Server to login to a server from remote computer. SSH uses 22/TCP port. OpenSSH is already installed by default even if you installed CentOS with "Minimal Install", so it's not necessary to install new packages. You can login with Password Authentication by default, but change some settings for security like follows.

```
[root@client ~]# -y install openssh (Already installed)
[root@dlp ~]# vi or nano /etc/ssh/sshd_config
# line 48: uncomment and change ( prohibit root login remotely )
PermitRootLogin no
# line 77: uncomment

PermitEmptyPasswords no
PasswordAuthentication yes
[root@dlp ~]# systemctl restart sshd
```

Configure SSH Client : CentOS

Configure SSH Client on CentOS.

Install SSH Client.

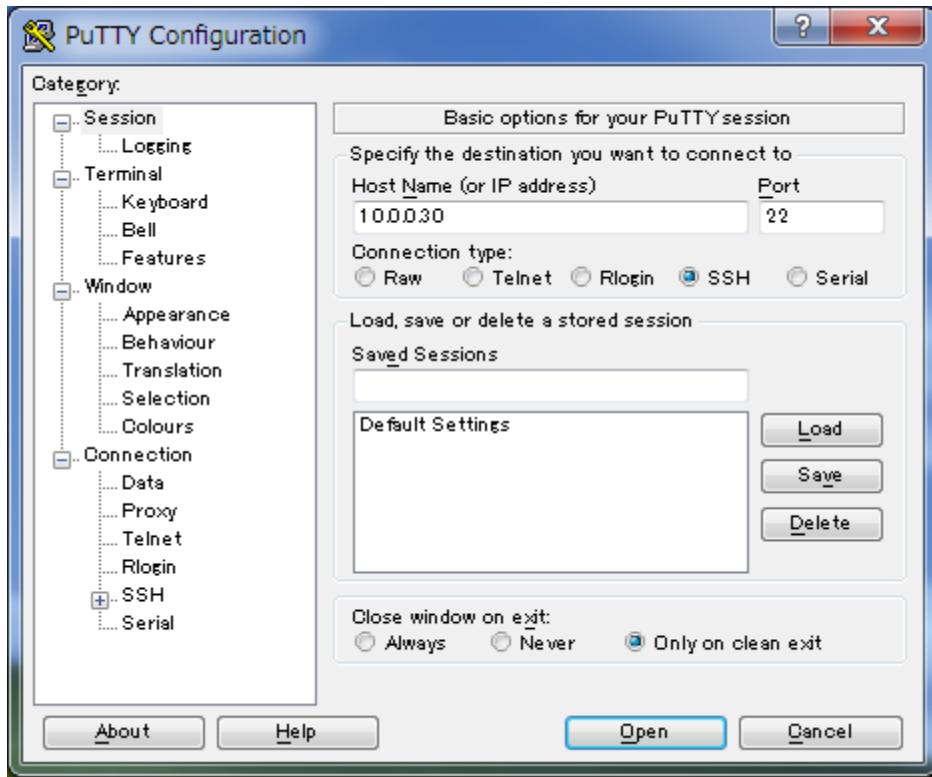
```
[root@client ~]# -y install openssh-clients
Connect to SSH server with a common user.
# ssh [username@(hostname or IP address)]
```

```
[root@client ~]# ssh cent@dlp.server.world
```

Configure SSH Client : Windows

Configure SSH Client on Windows.

Get a software which you can login with SSH from Windows clients. This example shows to use Putty. Install and start it and input your server's IP address and Click 'Open' button like follows.



[6] After succeding authentication, it's possible to login like follows.

A terminal window titled "cent@dlp:~" displays the following text:

```
login as: cent
cent@10.0.0.30's password:
Last login: Wed Jul  9 01:36:12 2014 from 10.0.0.5
[cent@dlp ~]$
```

The password input field is highlighted with a green rectangle.

Exporting Proxy

yum add proxy on CentOS:

http_proxy= http://172.16.200.1:3128

export http_proxy

yum update

There is a standard way to do it by **adding a proxy directive** to **/etc/yum.conf**

However for some reason:

```
proxy=http://your-proxy-url.com:8080
proxy_username=yum-user
proxy_password=qwerty
```

To make yum work via proxy in gnome-terminal run first:

```
export http_proxy=http://your-proxy-server.com:8080
```

or if proxy is protected by username / password run instead:

```
export username='yum-user'
export password='qwerty'
export http_proxy="http://$username:$password@your-proxy-server:8080/"
```

Afterwards yum will work via the proxy, i.e.:

```
yum update && yum upgrade
```

How to Install LAMP on CentOS 7

LAMP which originally stands for Linux, Apache, MySQL and PHP has now recently changed with the rise of [MariaDB](#), a drop-in replacement for original MySQL. Long story short, MariaDB is a fork of MySQL and developed by MySQL developers itself. It has almost all features of what MySQL has and features library binary equivalency and exact matching with MySQL APIs and commands. It means if an app is able to run with MySQL and it also is able to on MariaDB without any glitch. I will not explain what is [Apache and PHP](#) as I've explained before and I believe you already knew what it is.

You may need:

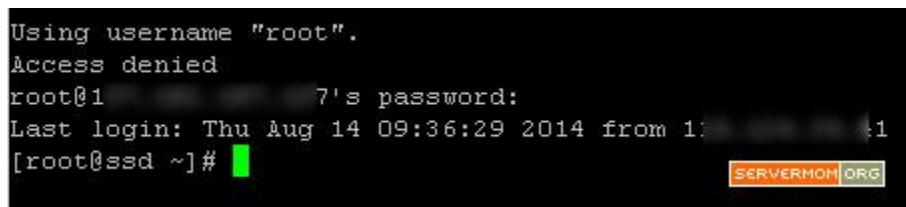
1. A server running CentOS 7. I recommend you to use CentOS 7 x86_64 minimal if available.
2. A knowledge on [how to use Putty or Terminal to access a server via SSH](#).
3. I believe you knew *-at least part of-* [most common Unix commands used to manage an unmanaged server](#).
4. A spare time of your life and a cup of coffee.

Install Apache Web Server

Being the most popular web server, Apache is commonly included in most of recent Linux Distro so installation will be very easy.

Before you proceed to the next steps, it is better to explain that all commands in this tutorial are written without the “sudo” prefix. In this tutorial I use root but you may also login as separate user with root privilege. However if you [disabled root login](#) and you logged in using another username with root privilege, you can add the “**sudo**” prefix all by your self. Alternatively you can simply type **su**, hit Enter and type in your password twice to switch as root.

Step 1 – Login to your server via Putty or Terminal.



```
Using username "root".
Access denied
root@1         7's password:
Last login: Thu Aug 14 09:36:29 2014 from 1
[root@ssd ~]#
```

Step 2 – Now issue command below to install Apache 2.4 on your CentOS 7 server:

```
yum install httpd -y
```

As you can see the command is still the same.



```
[root@ssd ~]# yum install httpd -y
Loaded plugins: fastestmirror
base
extras
updates
```

And when the process finished, you'll see something like this:

```
Installed:  
    httpd.x86_64 0:2.4.6-18.el7.centos  
  
Dependency Installed:  
    apr.x86_64 0:1.4.8-3.el7  
    centos-logos.noarch 0:70.0.6-1.el7.centos  
    mailcap.noarch 0:2.1.41-2.el7  
    apr-util.x86_64 0:1.5.1-1.el7  
    httpd-tools.x86_64 0:2.4.6-18.el7.centos  
  
Complete!  
[root@ssd ~]#
```

SERVERTOM|ORG

Step 3 – Now you have Apache 2.4 installed which you can then start the service by typing command below:

```
systemctl start httpd.service
```

or...

```
service httpd start
```

```
[root@ssd ~]# systemctl start httpd.service  
[root@ssd ~]#
```

SERVERTOM|ORG

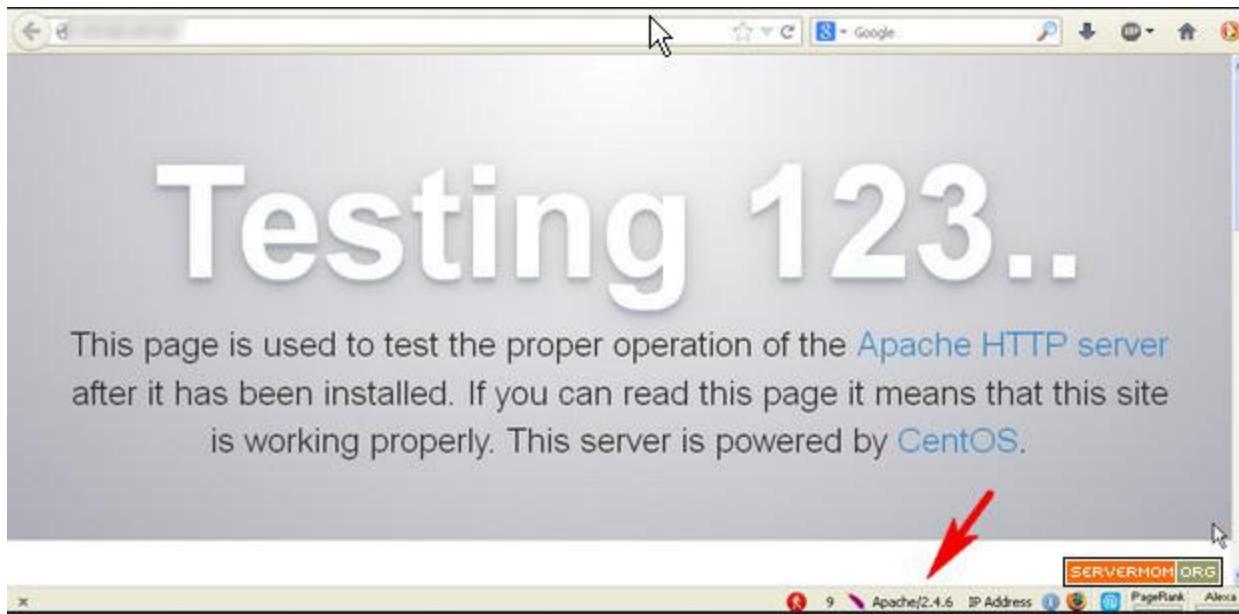
Available commands:

```
systemctl status|start|stop|restart|reload httpd.service  
# OR, old command :  
service httpd status|start|stop|restart|reload
```

Step 4 – You can verify that Apache is really running by opening your favorite web browser and access your vps via its IP address:

```
http://xxx.xxx.xxx.xxx
```

and you'll see default Apache welcome page.



In current example I've installed Apache v2.4.6.

or, you can directly issue this command:

```
systemctl status httpd.service
```

you'll see something like this:

```
[root@ssd ~]# systemctl status httpd.service
httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled)
   Active: active (running) since Fri 2014-08-15 11:58:08 EDT; 23min ago
     Main PID: 619 (httpd)
       Status: "Total requests: 9; Current requests/sec: 0; Current traffic: 0 B/sec"
      CGrouп: /system.slice/httpd.service
              ^a619 /usr/sbin/httpd -DFOREGROUND
              ^a620 /usr/sbin/httpd -DFOREGROUND
              ^a621 /usr/sbin/httpd -DFOREGROUND
              ^a622 /usr/sbin/httpd -DFOREGROUND
              ^a623 /usr/sbin/httpd -DFOREGROUND
              ^a624 /usr/sbin/httpd -DFOREGROUND
              ^a628 /usr/sbin/httpd -DFOREGROUND
              ^a629 /usr/sbin/httpd -DFOREGROUND
              ^a630 /usr/sbin/httpd -DFOREGROUND
              ^a631 /usr/sbin/httpd -DFOREGROUND
              ^a632 /usr/sbin/httpd -DFOREGROUND

Aug 15 11:58:08 ssd.servermom.org systemd[1]: Started The Apache HTTP Server.
[root@ssd ~]#
```

Enable Apache to automatically run every time your server reboot:

```
systemctl enable httpd.service
```

Install PHP5

Step 5 – Now, it is time to install PHP5. Default command is:

```
yum install php -y
```

That' really is a simple command but since we will install MySQL then we'll need PHP MySQL modules plus any other PHP5 modules you website / app may need it. You can view all available modules using this command:

```
yum search php-
```

Confused? You can read my [previous article](#) or you can simply use command below that includes common PHP5 modules most websites can run with it.

```
yum install php php-common php-cli php-devel php-gd php-imap php-intl php-mysql php-process php-xml php-xmlrpc php-zts -y
```

```
[root@ssd ~]# yum install php php-common php-cli php-devel php-gd php-imap php-intl php-mysql php-process php-xml php-xmlrpc php-zts -y
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: repos.dfw.quadranet.com
 * extras: mirrors.eeasynews.com
 * updates: mirrors.kernel.org
No package php-imap available.
No package php-zts available.
Resolving Dependencies
--> Running transaction check
--> Package php.x86_64 0:5.4.16-23.el7_0 will be installed
--> Package php-cli.x86_64 0:5.4.16-23.el7_0 will be installed
--> Package php-common.x86_64 0:5.4.16-23.el7_0 will be installed
--> Processing Dependency: libzip.so.2()(64bit) for package: php-common-5.4.16-23.el7_0.x86_64
--> Package php-devel.x86_64 0:5.4.16-23.el7_0 will be installed
--> Processing Dependency: pcre-devel(x86-64) for package: php-devel-5.4.16-23.el7_0.x86_64
--> Processing Dependency: automake for package: php-devel-5.4.16-23.el7_0.x86_64
```

And once done, you'll see something like this:

```
Installed:
  php.x86_64 0:5.4.16-23.el7_0
  php-common.x86_64 0:5.4.16-23.el7_0
  php-gd.x86_64 0:5.4.16-23.el7_0
  php-mysql.x86_64 0:5.4.16-23.el7_0
  php-xml.x86_64 0:5.4.16-23.el7_0

Dependency Installed:
  autoconf.noarch 0:2.69-11.el7
  freetype.x86_64 0:2.4.11-9.el7
  libX11-common.noarch 0:1.6.0-2.1.el7
  libXpm.x86_64 0:3.5.10-5.1.el7
  libjpeg-turbo.x86_64 0:1.2.90-5.el7
  libxcb.x86_64 0:1.9-5.el7
  libzip.x86_64 0:0.10.1-8.el7
  pcre-devel.x86_64 0:8.32-12.el7
  perl-Thread-Queue.noarch 0:3.02-2.el7
  tlibx.x86_64 0:5.1.2-14.el7

php-cli.x86_64 0:5.4.16-23.el7_0
php-devel.x86_64 0:5.4.16-23.el7_0
php-intl.x86_64 0:5.4.16-23.el7_0
php-process.x86_64 0:5.4.16-23.el7_0
php-xmirepc.x86_64 0:5.4.16-23.el7_0

automake.noarch 0:1.13.4-3.el7
libXiI.x86_64 0:1.6.0-2.1.el7
libXau.x86_64 0:1.0.8-2.1.el7
libicu.x86_64 0:50.1.2-11.el7
libpng.x86_64 2:1.5.13-5.el7
libXslt.x86_64 0:1.1.28-5.el7
m4.x86_64 0:1.4.16-9.el7
perl-Test-Harness.noarch 0:3.28-2.el7
php-pdo.x86_64 0:5.4.16-23.el7_0

Complete!
[root@ssd ~]#
```

SERVERTUTORIAL.ORG

You can test which version of PHP is installed by typing **php -v** command.

```
[root@ssd ~]# php -v
PHP 5.4.16 (cli) (built: Aug 6 2014 13:12:28)
Copyright (c) 1997-2013 The PHP Group
Zend Engine v2.4.0, Copyright (c) 1998-2013 Zend Technologies
[root@ssd ~]#
```

SERVERTUTORIAL.ORG

In my example it is **PHP v5.4.16**.

Install MariaDB MySQL Server

Step 6 – Installing MariaDB mysql server on CentOS 7 is pretty easy and once again we'll make us of yum package manager:

```
yum install mariadb-server mariadb -y
```

```
[root@ssd ~]# yum install mariadb-server mariadb -y
Loaded plugins: fastestmirror
base
extras
updates
.
```

SERVERTUTORIAL.ORG

and once done you'll see something like this:

```

Installing : perl-DBI-1.627-4.el7.x86_64                                     8
Installing : perl-DBD-MySQL-4.023-5.el7.x86_64                               9
Installing : 1:mariadb-server-5.5.37-1.el7_0.x86_64                           10
Verifying  : perl-Compress-Raw-Bzip2-2.061-3.el7.x86_64                         1
Verifying  : perl-Net-Daemon-0.48-5.el7.noarch                                2
Verifying  : 1:mariadb-5.5.37-1.el7_0.x86_64                               3
Verifying  : perl-PIRPC-0.2020-14.el7.noarch                                4
Verifying  : 1:mariadb-server-5.5.37-1.el7_0.x86_64                           5
Verifying  : 1:perl-Compress-Raw-Zlib-2.061-4.el7.x86_64                      6
Verifying  : libaio-0.3.109-12.el7.x86_64                                 7
Verifying  : perl-DBI-1.627-4.el7.x86_64                                     8
Verifying  : perl-IO-Compress-2.061-2.el7.noarch                            9
Verifying  : perl-DBD-MySQL-4.023-5.el7.x86_64                           10

Installed:
  mariadb.x86_64 1:5.5.37-1.el7_0                                         mariadb-server.x86_64 1:5.5.37-1.el7_0

Dependency Installed:
  libaio.x86_64 0:0.3.109-12.el7                                              perl-Compress-Raw-Bzip2.x86_64 0:2.061-3.el7
  perl-Compress-Raw-Zlib.x86_64 1:2.061-4.el7                                perl-DBD-MySQL.x86_64 0:4.023-5.el7
  perl-DBI.x86_64 0:1.627-4.el7                                              perl-IO-Compress.noarch 0:2.061-2.el7
  perl-Net-Daemon.noarch 0:0.48-5.el7                                         perl-PIRPC.noarch 0:0.2020-14.el7

Complete!
[root@ssd ~]# [REDACTED]

```

Step 7 – Now you can start MariaDB server for the very first time using this simple systemctl command :

```
systemctl start mariadb.service
```

```

[root@ssd ~]# systemctl start mariadb.service
[root@ssd ~]# systemctl status mariadb
mariadb.service - MariaDB database server
  Loaded: loaded (/usr/lib/systemd/system/mariadb.service; disabled)
  Active: active (running) since Sat 2014-08-16 10:07:10 EDT; 38s ago
    Process: 954 ExecStartPost=/usr/libexec/mariadb-wait-ready $MAINPID
)
   Process: 875 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (c
 Main PID: 953 (mysqld_safe)
   CGroup: /system.slice/mariadb.service
           ââ 953 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
           ââ1109 /usr/libexec/mysqld --basedir=/usr --datadir=/var/l

Aug 16 10:07:08 ssd.servermom.org mariadb-prepare-db-dir[875]: The la
Aug 16 10:07:08 ssd.servermom.org mariadb-prepare-db-dir[875]: You ca
Aug 16 10:07:08 ssd.servermom.org mariadb-prepare-db-dir[875]: http://
Aug 16 10:07:08 ssd.servermom.org mariadb-prepare-db-dir[875]: Suppor
Aug 16 10:07:08 ssd.servermom.org mariadb-prepare-db-dir[875]: SkySQL
Aug 16 10:07:08 ssd.servermom.org mariadb-prepare-db-dir[875]: Altern
Aug 16 10:07:08 ssd.servermom.org mariadb-prepare-db-dir[875]: http://
Aug 16 10:07:08 ssd.servermom.org mysqld_safe[953]: 140816 10:07:08 m
Aug 16 10:07:08 ssd.servermom.org mysqld_safe[953]: 140816 10:07:08 m
Aug 16 10:07:10 ssd.servermom.org systemd[1]: Started MariaDB databases.
Hint: Some lines were ellipsized, use -l to show in full. [REDACTED]
[root@ssd ~]# [REDACTED]

```

You may also see the status of MariaDB by typing:

```
systemctl status mariadb
```

MariaDB Initial Configuration

Step 8 – So its service is now running but there is one thing you should do immediately: configuring MariaDB setup for the very first time like setting up your mysql root password. Issue this command:

```
mysql_secure_installation
```

Then you'll see a series of question, just answer it accordingly. The main important part is to define your root password while everything else is just up to you or you can simply hit the "ENTER" key through each prompt to accept the default values.

```
[root@ssd ~]# mysql_secure_installation
/usr/bin/mysql_secure_installation: line 379: find_mysql_client: command
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user.  If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

Set root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] n
... skipping.

By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
... Success!

Cleaning up...

All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
[root@ssd ~]#
```

If you need to automatically run MariaDB everytime your server boot, simply issue this command:

```
systemctl enable mariadb.service
```

```
[root@ssd ~]# systemctl enable mariadb.service
ln -s '/usr/lib/systemd/system/mariadb.service' '/etc/systemd/system/multi-user.target.wants/mariadb.service'
[root@ssd ~]#
```

SERVERTMOM.ORG

You may also need to test your newly installed MariaDB by logging in as root:

```
mysql -u root -p
```

```
[root@ssd ~]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 9
Server version: 5.5.37-MariaDB MariaDB Server

Copyright (c) 2000, 2014, Oracle, Monty Program Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
MariaDB [(none)]>
```

SERVERTMOM.ORG

As you can see from the screenshot above, it is **Maria DB v5.5.37**.

Step 9 – Also test if Apache and PHP is running well and able to process any *.php files. Create a php info page using this command followed by restarting apache

```
echo "<?php phpinfo(); ?>" > /var/www/html/info.php
```

```
[root@ssd ~]# echo "<?php phpinfo(); ?>" > /var/www/html/info.php
[root@ssd ~]# cd /var/www/html
[root@ssd html]# ls
info.php
```

SERVERTMOM.ORG

Restart apache:

```
systemctl restart httpd.service
```

Now open up your browser and access that newly created php page:

```
http://xxx.xxx.xxx.xxx/info.php
```

You'll see a page similar to this one:

System	Linux ssd.servermom.org 2.6.32-042stab092.3 #1 SMP Sun Jul 20 13:27:24 MSK 2014 x86_64
Build Date	Aug 6 2014 13:13:24
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/curl.ini, /etc/php.d/dom.ini, /etc/php.d/fileinfo.ini, /etc/php.d/gd.ini, /etc/php.d/intl.ini, /etc/php.d/json.ini, /etc/php.d/mysql.ini, /etc/php.d/mysqli.ini, /etc/php.d/pdo.ini, /etc/php.d/pdo_mysql.ini, /etc/php.d/pdo_sqlite.ini, /etc/php.d/phar.ini, /etc/php.d posix.ini, /etc/php.d/sqlite3.ini, /etc/php.d/sysvmsg.ini, /etc/php.d/sysvsem.ini, /etc/php.d/sysvshm.ini, /etc/php.d/wddx.ini, /etc/php.d/xmlreader.ini, /etc/php.d/xmlrpc.ini, /etc/php.d/xmlwriter.ini, /etc/php.d/xsl.ini, /etc/php.d/zip.ini

SERVERMOM.ORG

Installing PhpMyAdmin (see another method if this method not work for you)

Step 10 – Now your server has Apache, PHP and MariaDB installed. It means it should be OK now to install PhpMyAdmin, a popular web-based database management system so you can easily manage your database without having to login via SSH and issuing several command lines. Unluckily, this piece of awesome software is not available in CentOS 7.0 default repositories. In this case you have to add / enable third-party repo like **EPEL** or **RPMForge**.

Method #1: RPMForge

First, download the rpm file.

```
wget http://pkgs.repoforge.org/rpmforge-release/rpmforge-release-0.5.3-1.el7.rf.x86_64.rpm
```

```
[root@ssd ~]# yum http://pkgs.repoforge.org/rpmforge-release/rpmforge-release-0.5.3-1.el7.rf.x86_64.rpm
Loaded plugins: fastestmirror
No such command: http://pkgs.repoforge.org/rpmforge-release/rpmforge-release-0.5.3-1.el7.rf.x86_64.rpm. Please use /usr/bin/yum --help
[root@ssd ~]# wget http://pkgs.repoforge.org/rpmforge-release/rpmforge-release-0.5.3-1.el7.rf.x86_64.rpm
--2014-08-16 11:22:14-- http://pkgs.repoforge.org/rpmforge-release/rpmforge-release-0.5.3-1.el7.rf.x86_64.rpm
Resolving pkgs.repoforge.org (pkgs.repoforge.org)... 78.46.17.228
Connecting to pkgs.repoforge.org (pkgs.repoforge.org)|78.46.17.228|:80... connected.
HTTP request sent, awaiting response... 302 Found
Location: http://rpmforge.sw.be/redhat/el7/en/x86_64/rpmforge/RPMS/rpmforge-release-0.5.3-1.el7.rf.x86_64.rpm [following]
--2014-08-16 11:22:14-- http://rpmforge.sw.be/redhat/el7/en/x86_64/rpmforge/RPMS/rpmforge-release-0.5.3-1.el7.rf.x86_64.rpm
Resolving rpmforge.sw.be (rpmforge.sw.be)... 78.46.17.228
Connecting to rpmforge.sw.be (rpmforge.sw.be)|78.46.17.228|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://tree.repoforge.org/redhat/el7/en/x86_64/rpmforge/RPMS/rpmforge-release-0.5.3-1.el7.rf.x86_64.rpm [following]
```

then enable the repository and delete the .rpm file as it is not needed again.

```
rpm -ivh rpmforge-release-*
rm rpmforge-release-*
```

```
[root@ssd ~]# rpm -ivh rpmforge-release-*
warning: rpmforge-release-0.5.3-1.el7.rf.x86_64.rpm: Header V3 DSA/SHA1
NOKEY
Preparing... #####################################################
Updating / installing...
 1:rpmforge-release-0.5.3-1.el7.rf  #####
[root@ssd ~]# rm rpmforge-release-*
rm: remove regular file 'rpmforge-release-0.5.3-1.el7.rf.x86_64.rpm'
[root@ssd ~]#
```

Method #2: EPEL

Download the .rpm file:

```
wget http://download.fedoraproject.org/pub/epel/beta/7/x86_64/epel-release-7-0.2.noarch.rpm
```

The url above is still its beta version. However if the repository is out of beta status, the link most likely will be different. In case that happens, you can find out its latest download url at [Fedora Project website](#).

enable the repository and delete the .rpm file:

```
rpm -ivh epel-release*
rm epel-release*
```

Step 11 – Next, install it using yum again:

```
yum install phpmyadmin -y
```

screenshot:

```
[root@ssd ~]# rpm -ivh epel-release*
warning: epel-release-7-0.2.noarch.rpm: Header V3 RSA/SHA256 Signature, key ID
Preparing... ################################################ [100%]
Updating / installing...
1:epel-release-7-0.2 ################################################ [100%]
[root@ssd ~]# yum install phpmyadmin -y
Loaded plugins: fastestmirror
epel/x86_64/metalink
epel
(1/2): epel/x86_64/group_gz
(2/2): epel/x86_64/primary_db
Loading mirror speeds from cached hostfile
 * base: repos.dfw.quadranet.com
 * epel: linux.mirrors.es.net
 * extras: mirrors.easynews.com
 * rpmforge: mirror.hmc.edu
 * updates: mirrors.kernel.org
Resolving Dependencies
--> Running transaction check
--> Package phpMyAdmin.noarch 0:4.2.7-1.el7 will be installed
--> Processing Dependency: php-mcrypt >= 5.3.0 for package: phpMyAdmin-4.2.7-
--> Processing Dependency: php-mbstring >= 5.3.0 for package: phpMyAdmin-4.2.7-
--> Processing Dependency: php-tcpdf-dejavu-sans-fonts for package: phpMyAdmin-
--> Processing Dependency: php-tcpdf for package: phpMyAdmin-4.2.7-1.noarch
--> Processing Dependency: php-php-gettext for package: phpMyAdmin-4.2.7-1.noarch
[SERVERTOMORG]
```

Step 12 – That's it. Now you also have phpMyAdmin (PMA) installed but you should be better if you change its default configuration before using it. First, you'll need to backup default PMA's config file:

```
cp /etc/httpd/conf.d/phpMyAdmin.conf /etc/httpd/conf.d/phpMyAdmin.conf.old
```

then edit file phpMyAdmin.conf file using your favorite editor. In this example I use Nano editor:

```
nano /etc/httpd/conf.d/phpMyAdmin.conf
```

Step 13 – You'll now see the content of phpMyAdmin.conf, next you have to allow connections from remote hosts by editing few lines inside section `<Directory "/usr/share/phpMyAdmin">`.

Before changes:

```
<Directory /usr/share/phpMyAdmin/>
<IfModule mod_authz_core.c>
    # Apache 2.4
    <RequireAny>
        Require ip 127.0.0.1
        Require ip ::1
    </RequireAny>
</IfModule>
```

After:

```
<Directory /usr/share/phpMyAdmin/>
<IfModule mod_authz_core.c>
# Apache 2.4
<RequireAny>
    Require all granted
</RequireAny>
</IfModule>
<IfModule !mod_authz_core.c> SERVERMOM|ORG
# Apache 2.2
```

Also you'll need to edit few lines next:

Before:

```
# Apache 2.4
<RequireAny>
    Require ip all granted
</RequireAny>
</IfModule>
<IfModule !mod_authz_core.c>
# Apache 2.2
Order Deny,Allow
Deny from All
Allow from 127.0.0.1
Allow from ::1
</IfModule>
</Directory>
```

After:

```
# Apache 2.4
<RequireAny>
    Require ip all granted
</RequireAny>
</IfModule>
<IfModule !mod_authz_core.c>
# Apache 2.2
#Order Deny,Allow
#Deny from All
AllowOverride None
Options None
Allow from All
Require all granted
</IfModule>
</Directory>
```

Shortly it should look like this:

```
<Directory /usr/share/phpMyAdmin/>

<IfModule mod_authz_core.c>

# Apache 2.4

<RequireAny>
```

```
    Require all granted

</RequireAny>

</IfModule>

<IfModule !mod_authz_core.c>
    # Apache 2.2

    #Order Deny,Allow

    #Deny from All

    AllowOverride None

    Options None

    Allow from All

    Require all granted

</IfModule>

</Directory>
```

Once done, save and exit editor (In Nano it is Control+O then Control+X).

Step 14 – Restart Apache again:

```
systemctl restart httpd.service
```

Now you can test opening PMA on your browser via your server's IP address:

<http://xxx.xxx.xxx.xxx/phpmyadmin>

and default login page of phpMyAdmin should be displayed:



That's all. Now you can host your websites or blogs in that server, even WordPress.

Do not forget to follow me on [twitter](#) to get faster update or [download my official Android app](#). Enjoy..

phpMyAdmin configuration working methods

How To Install and Secure phpMyAdmin with Apache on a CentOS 7 Server

Introduction

Relational database management systems like MySQL and MariaDB are needed for a significant portion of web sites and applications. However, not all users feel comfortable administering their data from the command line.

To solve this problem, a project called phpMyAdmin was created in order to offer an alternative in the form of a web-based management interface. In this guide, we will demonstrate how to install and secure a phpMyAdmin configuration on a CentOS 7 server. We will build this setup on top of the Apache web server, the most popular web server in the world.

Prerequisites

Before we begin, there are a few requirements that need to be settled.

To ensure that you have a solid base to build this system upon, you should run through our [initial server setup guide for CentOS 7](#). Among other things, this will walk you through setting up a non-root user with `sudo` access for administrative commands.

The second prerequisite that must be fulfilled in order to start on this guide is to install a LAMP (Linux, Apache, MariaDB, and PHP) stack on your CentOS 7 server. This is the platform that we will use to serve our phpMyAdmin interface (MariaDB is also the database management software that we are wishing to manage). If you do not yet have a LAMP installation on your server, follow our tutorial on [installing LAMP on CentOS 7](#).

When your server is in a properly functioning state after following these guides, you can continue on with the rest of this page.

Step One — Install phpMyAdmin

With our LAMP platform already in place, we can begin right away with installing the phpMyAdmin software. Unfortunately, phpMyAdmin is not available in CentOS 7's default repository.

To get the packages we need, we'll have to add an additional repo to our system. The EPEL repo (**E**xtra **P**ackages for **E**nterprise **L**inux) contains many additional packages, including the phpMyAdmin package we are looking for.

The EPEL repository can be made available to your server by installing a special package called `epel-release`. This will reconfigure your repository list and give you access to the EPEL packages.

To install, just type:

```
sudo yum install epel-release
```

Now that the EPEL repo is configured, you can install the phpMyAdmin package using the `yum` packaging system by typing:

```
sudo yum install phpmyadmin
```

The installation will now complete. The installation included an Apache configuration file that has already been put into place. We will need to modify this a bit to get it to work correctly for our installation.

Open the file in your text editor now so that we can make a few changes:

```
sudo nano /etc/httpd/conf.d/phpMyAdmin.conf
```

Inside, we see some directory blocks with some conditional logic to explain the access policy for our directory. There are two distinct directories that are defined, and within these, configurations that will be valid for both Apache 2.2 and Apache 2.4 (which we are running).

Currently, this setup is configured to deny access to any connection not being made from the server itself. Since we are working on our server remotely, we need to modify some lines to specify the IP address of your *home* connection.

Change any lines that read `Require ip 127.0.0.1` or `Allow from 127.0.0.1` to refer to your home connection's IP address. If you need help finding the IP address of your home connection, check out the next section. There should be four locations in the file that must be changed:

```
...
Require ip your_workstation_IP_address
...
Allow from your_workstation_IP_address
...
Require ip your_workstation_IP_address
...
Allow from your_workstation_IP_address
...
```

When you are finished, restart the Apache web server to implement your modifications by typing:

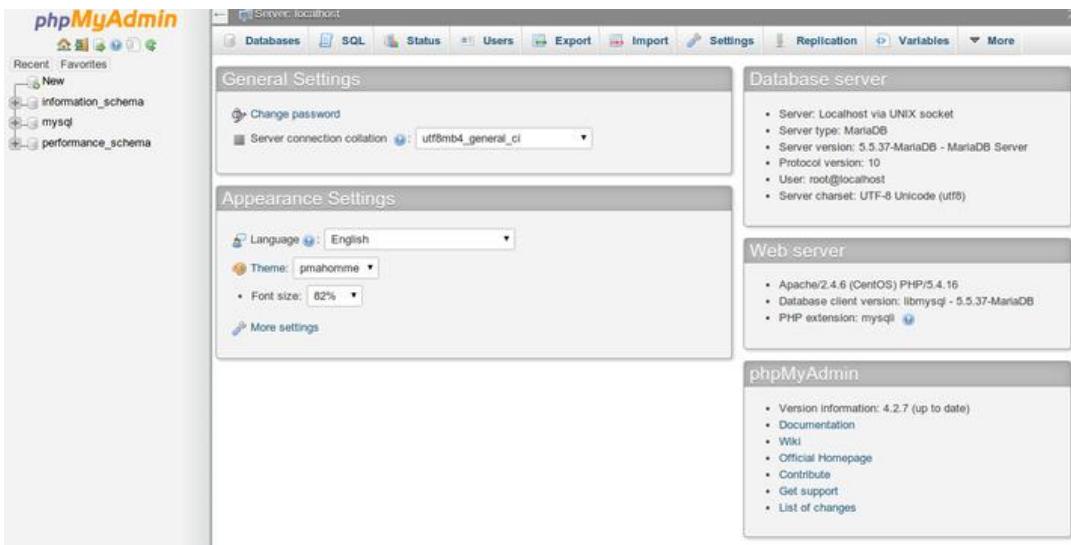
```
sudo systemctl restart httpd.service
```

With that, our phpMyAdmin installation is now operational. To access the interface, go to your server's domain name or public IP address followed by `/phpMyAdmin`, in your web browser:

```
http://server_domain_or_IP/phpMyAdmin
```



To sign in, use a username/password pair of a valid MariaDB user. The `root` user and the MariaDB administrative password is a good choice to get started. You will then be able to access the administrative interface:



Find Your IP Address

You will need to know the IP address of the computer you are using to access your databases in order to complete the step above. This is a security precaution so that unauthorized people cannot connect to your server.

Note: This is *not* the IP address of your VPS, it is the IP address of your home or work computer.

You can find out how the greater web sees your IP address by visiting one of these sites in your web browser:

- [What's My IP Address?](#)
- [What's My IP?](#)
- [My IP Address](#)

Compare a few different sites and make sure they all give you the same value. Use this value in the configuration file above.

Step Two — Secure your phpMyAdmin Instance

The phpMyAdmin instance installed on our server should be completely usable at this point. However, by installing a web interface, we have exposed our MySQL system to the outside world.

Even with the included authentication screen, this is quite a problem. Because of phpMyAdmin's popularity combined with the large amount of data it provides access to, installations like these are common targets for attackers.

We will implement two simple strategies to lessen the chances of our installation being targeted and compromised. We will change the location of the interface from /phpMyAdmin to something else to sidestep some of the automated bot brute-force attempts. We will also create an additional, web server-level authentication gateway that must be passed before even getting to the phpMyAdmin login screen.

Changing the Application's Access Location

In order for our Apache web server to work with phpMyAdmin, our phpMyAdmin Apache configuration file uses an alias to point to the directory location of the files.

To change the URL where our phpMyAdmin interface can be accessed, we simply need to rename the alias. Open the phpMyAdmin Apache configuration file now:

```
sudo nano /etc/httpd/conf.d/phpMyAdmin.conf
```

Toward the top of the file, you will see two lines that look like this:

```
Alias /phpMyAdmin /usr/share/phpMyAdmin  
Alias /phpmyadmin /usr/share/phpMyAdmin
```

These two lines are our aliases, which means that if we access our site's domain name or IP address, followed by either `/phpMyAdmin` or `/phpmyadmin`, we will be served the content at `/usr/share/phpMyAdmin`.

We want to disable these specific aliases since they are heavily targeted by bots and malicious users. Instead, we should decide on our own alias. It should be easy to remember, but not easy to guess. It shouldn't indicate the purpose of the URL location. In our case, we'll go with `/nothingtosee`.

To apply our intended changes, we should remove or comment out the existing lines and add our own:

```
# Alias /phpMyAdmin /usr/share/phpMyAdmin  
# Alias /phpmyadmin /usr/share/phpMyAdmin  
Alias /nothingtosee /usr/share/phpMyAdmin
```

When you are finished, save and close the file.

To implement the changes, restart the web service:

```
sudo systemctl restart httpd.service
```

Now, if you go to the previous location of your phpMyAdmin installation, you will get a 404 error:

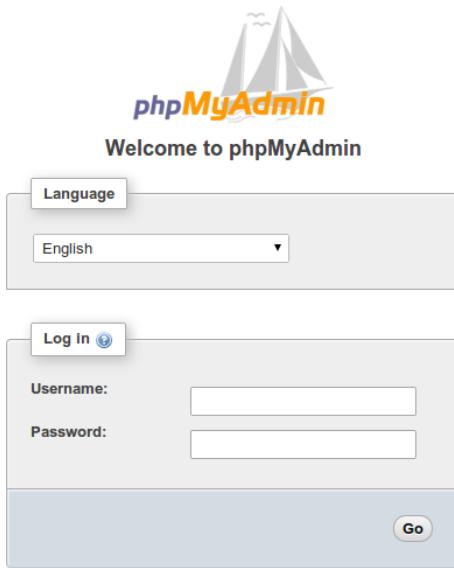
```
http://server_domain_or_IP/phpMyAdmin
```

Not Found

The requested URL `/phpMyAdmin` was not found on this server.

However, your phpMyAdmin interface will be available at the new location we selected:

```
http://server_domain_or_IP/nothingtosee
```



Setting up a Web Server Authentication Gate

The next feature we wanted for our installation was an authentication prompt that a user would be required to pass before ever seeing the phpMyAdmin login screen.

Fortunately, most web servers, including Apache, provide this capability natively. We will just need to modify our Apache configuration file to use an authorization file.

Open the phpMyAdmin Apache configuration file in your text editor again:

```
sudo nano /etc/httpd/conf.d/phpMyAdmin.conf
```

Within the `/usr/share/phpMyAdmin` directory block, but outside of any of the blocks inside, we need to add an override directive. It will look like this:

```
...
<Directory /usr/share/phpMyAdmin/>
    AllowOverride All
    <IfModule mod_authz_core.c>
        .
        .
    </IfModule>
    .
.
```

This will allow us to specify additional configuration details in a file called `.htaccess` located *within* the phpMyAdmin directory itself. We will use this file to set up our password authentication.

Save and close the file when you are finished.

Restart the web service to implement this change:

```
sudo systemctl restart httpd.service
```

Create an .htaccess File

Now that we have the override directive in our configuration, Apache will look for a file called `.htaccess` within the `/usr/share/phpMyAdmin` directory. If it finds one, it will use the directives contained within to supplement its previous configuration data.

Our next step is to create the `.htaccess` file within that directory. Use your text editor to do so now:

```
sudo nano /usr/share/phpMyAdmin/.htaccess
```

Within this file, we need to enter the following information:

```
AuthType Basic  
AuthName "Admin Login"  
AuthUserFile /etc/httpd/pma_pass  
Require valid-user
```

Let's go over what each of these lines mean:

- **AuthType Basic:** This line specifies the authentication type that we are implementing. This type will implement password authentication using a password file.
- **AuthName:** This sets the message for the authentication dialog box. You should keep this generic so that unauthorized users won't gain knowledge about what is being protected.
- **AuthUserFile:** This sets the location of the actual password file that will be used for authentication. This should be outside of the directories that are being served. We will create this file in a moment.
- **Require valid-user:** This specifies that only authenticated users should be given access to this resource. This is what actually stops unauthorized users from entering.

When you are finished entering this information, save and close the file.

Create the Password File for Authentication

Now that we have specified the location for our password file through the use of the `AuthUserFile` directive in our `.htaccess` file, we need to create and populate the password file.

This can be accomplished through the use of an Apache utility called `htpasswd`. We invoke the command by passing it the location where we would like to create the file and the username we would like to enter authentication details for:

```
sudo htpasswd -c /etc/httpd/pma_pass username
```

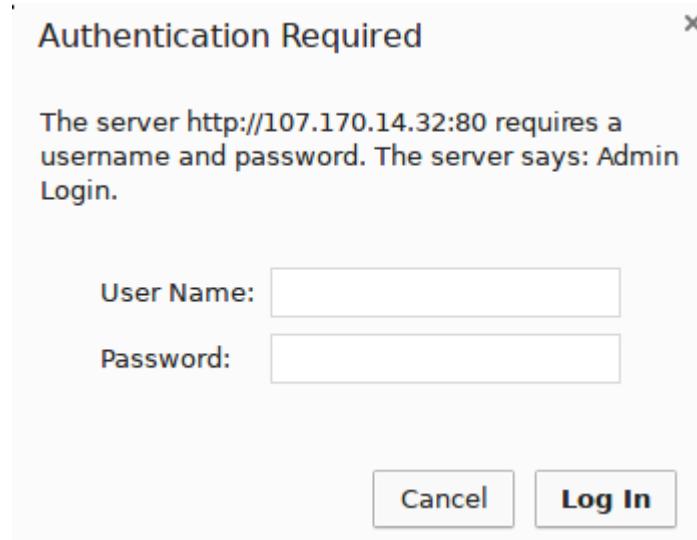
The `-c` flag indicates that this will create an initial file. The directory location is the path and filename that will be used for the file. The username is the first user we would like to add. You will be prompted to enter and confirm a password for the user.

If you want to add additional users to authenticate, you can call the same command again **without** the `-c` flag, and with a new username:

```
sudo htpasswd /etc/httpd/pma_pass seconduser
```

With our password file created, an authentication gateway has been implemented and we should now see a password prompt the next time we visit our site:

`http://server_domain_or_IP/nothingtosee`



Once you enter your credentials, you will be taken to the normal phpMyAdmin login page. This added layer of protection will help keep your MySQL logs clean of authentication attempts in addition to the added security benefit.

Conclusion

You can now manage your MySQL databases from a reasonably secure web interface. This UI exposes most of the functionality that is available from the MySQL command prompt. You can view databases and schema, execute queries, and create new data sets and structures.

The configuration file will be like that

```
Alias /phpMyAdmin /usr/share/phpMyAdmin  
Alias /phpmyadmin /usr/share/phpMyAdmin
```

```
<Directory /usr/share/phpMyAdmin/>  
    AddDefaultCharset UTF-8  
  
<IfModule mod_authz_core.c>  
    # Apache 2.4  
    <RequireAny>  
    # Require ip 172.16.151.15  
    # Require ip 172.16.151.26  
    # Require ip ::1
```

```
    Require all granted
</RequireAny>
</IfModule>
<IfModule !mod_authz_core.c>
# Apache 2.2
Order Deny,Allow
# Deny from All
Allow from 172.16.151.15
Allow from 172.16.151.26
Allow from ::1
</IfModule>
</Directory>

<Directory /usr/share/phpMyAdmin/setup/>
<IfModule mod_authz_core.c>
# Apache 2.4
<RequireAny>
Require ip 172.16.151.15
Require ip 172.16.151.26
Require ip ::1
</RequireAny>
</IfModule>
<IfModule !mod_authz_core.c>
# Apache 2.2
Order Deny,Allow
# Deny from All
Allow from 172.16.151.15
Allow from 172.16.151.26
Allow from ::1
</IfModule>
</Directory>

# These directories do not require access over HTTP - taken from the original
# phpMyAdmin upstream tarball
#
<Directory /usr/share/phpMyAdmin/libraries/>
Order Deny,Allow
# Deny from All
# Allow from None
Allow from all
</Directory>

<Directory /usr/share/phpMyAdmin/setup/lib/>
# Order Deny,Allow
# Deny from All
# Allow from None
```

```
Allow from all
</Directory>

<Directory /usr/share/phpMyAdmin/setup/frames/>
    Order Deny,Allow
    # Deny from All
    # Allow from None
    Allow from all
</Directory>

# This configuration prevents mod_security at phpMyAdmin directories from
# filtering SQL etc. This may break your mod_security implementation.
#
#<IfModule mod_security.c>
#  <Directory /usr/share/phpMyAdmin/>
#    SecRuleInheritance Off
#  </Directory>
#</IfModule>
```

Now install WordPress.....

Happy Server Configuration

Engr. Md. Nazim Uddin

nazim.cse.kuet@gmail.com

Changing from 32 to 64bit

- If your Virtual Box only showing 32 bit =

<http://www.fixedbyvonne.com/2014/11/virtualbox-showing-32-bit-guest-versions-64-bit-host-os/>

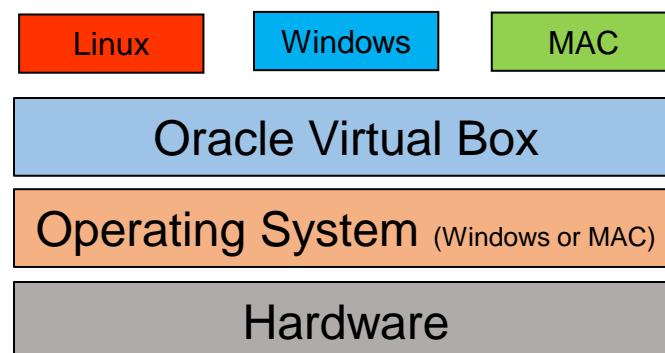
- Help to release **mouse** capture in **virtualBox**. Press one key by itself: *The [CTRL] key on the right hand side of the keyboard

WELCOME TO: MODULE 2

**DOWNLOAD, INSTALL AND
CONFIGURE LINUX**

What is Oracle VirtualBox?

- VirtualBox is a free and open-source hypervisor for x86 computers currently being developed by Oracle Corporation
- It installs on your existing Intel or AMD-based computers, whether they are running Windows, Mac, Linux or Solaris operating systems. It extends the capabilities of your existing computer so that it can run multiple operating systems on one hardware at the same time



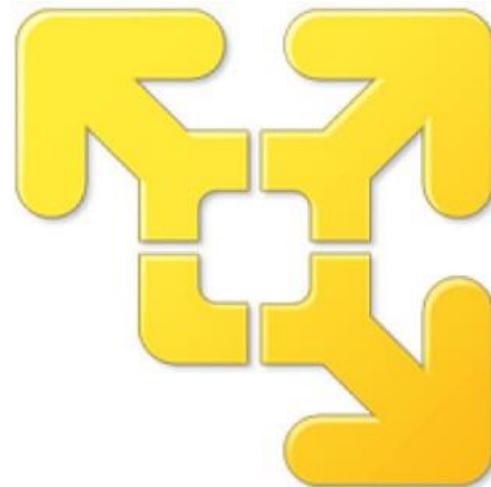
ORACLE VIRTUAL BOX DOWNLOAD AND INSTALLATION



CREATE VIRTUAL MACHINE

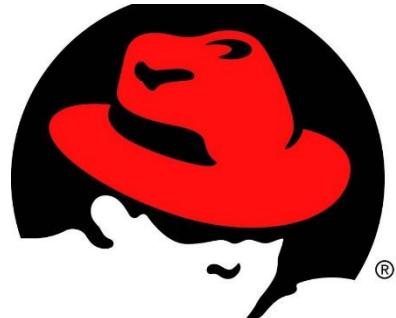


VMWARE WORKSTATION PLAYER



Download, Installation and Configuration

LINUX DISTRIBUTIONS

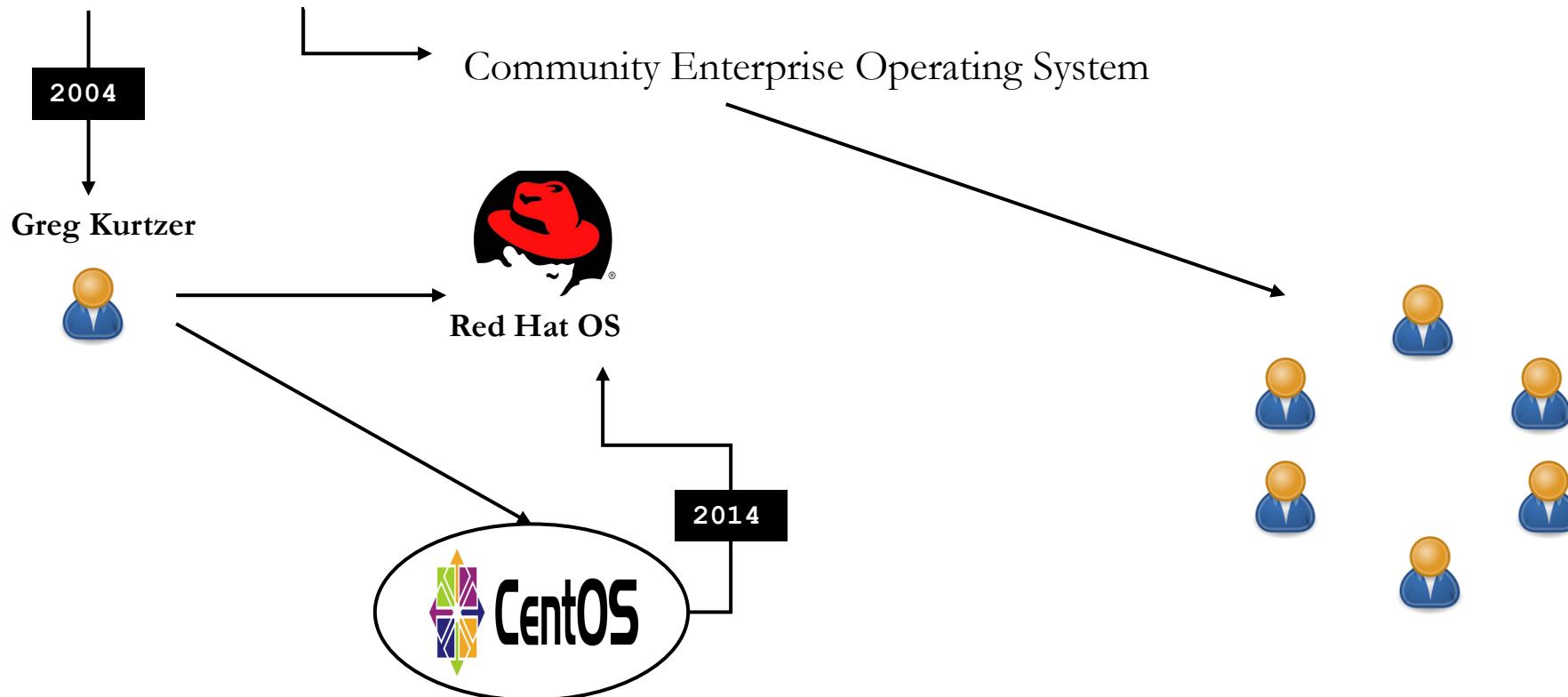


Redhat



CentOS vs. CentOS Stream

- Brief history of CentOS



CentOS vs. CentOS Stream

Before Feb 2021

Fedora → RHEL → CentOS

After Feb 2021

Fedora → CentOS Stream → RHEL

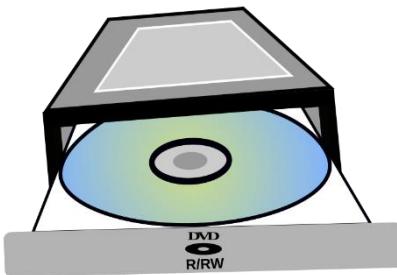
CentOS vs. CentOS Stream

Question???

Is it worth learning CentOS

*** **ABSOLUTELY** ***

DIFFERENT WAYS TO INSTALL OS



CENTOS INSTALLATION (VERSION 7)

Recommended



CENTOS INSTALLATION (VERSION 8)



Linux Download and Installation



CentOS Installation 8



NEXT LESSON

REDHAT LINUX INSTALLATION



OPTIONAL

REDHAT LINUX INSTALLATION





VirtualBox SNAPSHOTS

LINUX DESKTOP GUI

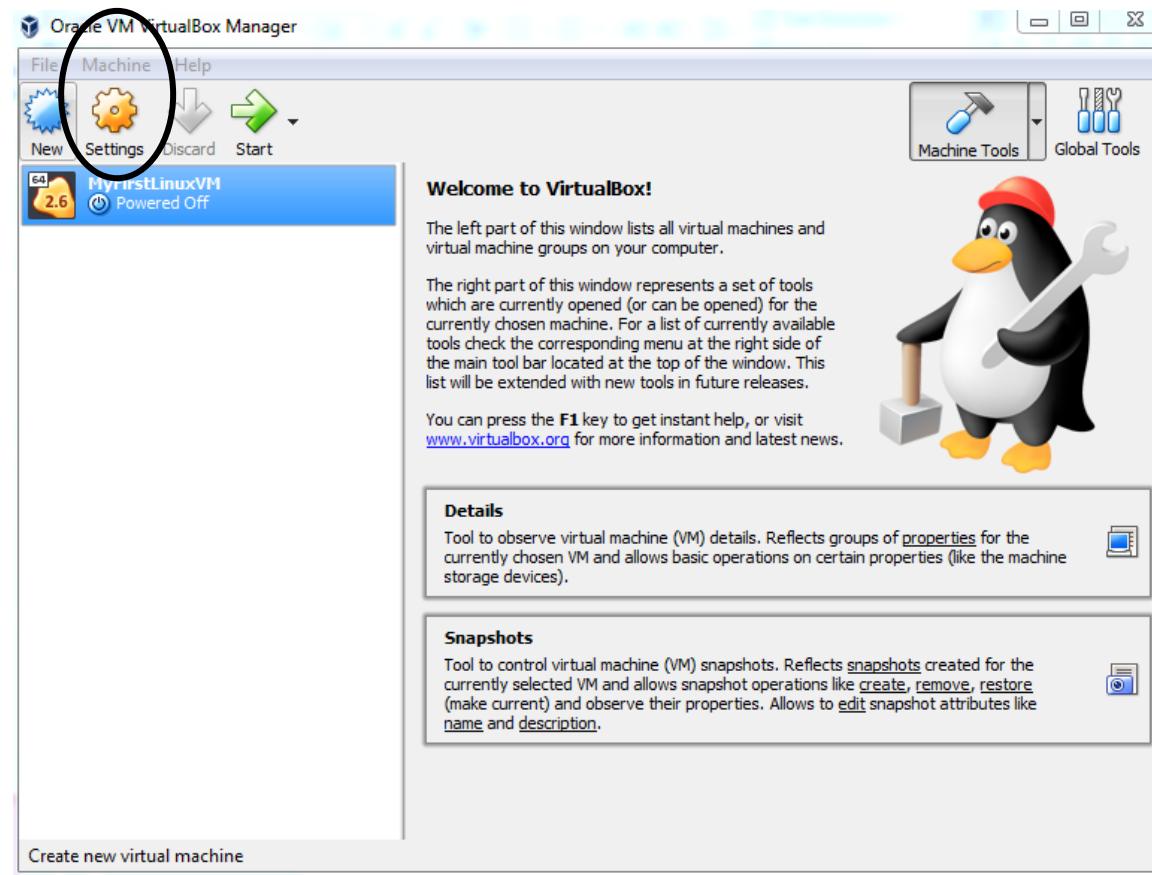
(GNOME OR KDE)

LINUX UBUNTU INSTALLATION



OPTIONAL

ORACLE VIRTUAL MACHINE MANAGEMENT



LINUX VS. WINDOWS

	Linux	Windows
Price	Free	\$\$\$
Ease	Not user-friendly	User friendly
Reliability	Very reliable. Often runs for months or years	Often requires reboot
Software	Mostly enterprise level softwares	Much larger selection of softwares e.g. office, games, utilities etc.
Multi-tasking	Best for multi-tasking	Multi-tasking is available but with very high cpu or memory resources
Security	Very secure	Some what secure
Open source	Open to public	No an open source OS

LINUX USERS

- US Government and Agencies (National, State, Federal and International)
- NASA
- Health Care
- The Bullet Trains in Japan runs at the speed of 150-215 m/h
- Traffic Control
- Financial Institutes e.g. NYSE
- Entertainment industries (Cinemas, Production houses etc.)
- World e-commerce leaders, including Amazon, eBay, PayPal, and Walmart run their platform on Linux
- Other fortune 500 companies e.g. Google, IBM, McDonalds, Facebook etc.

Keyboard Keys Used in Linux



Right Ctrl key

HANDOUTS

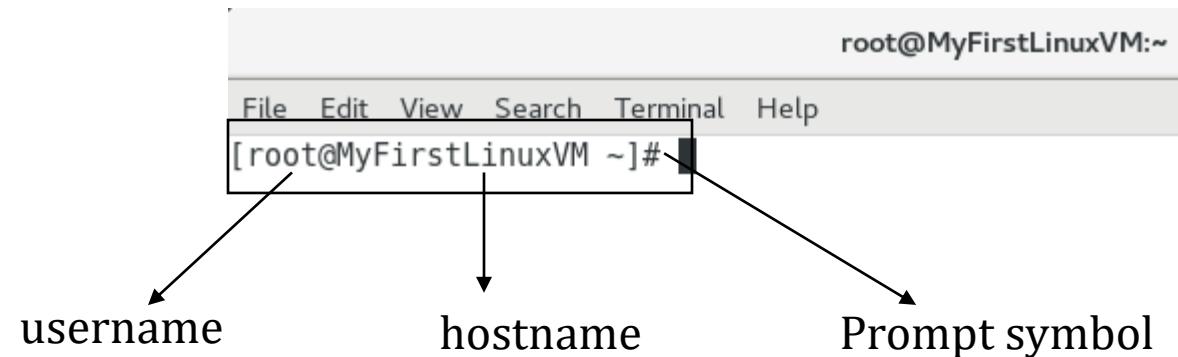
MODULE 3

WELCOME TO: MODULE 3

SYSTEM ACCESS AND FILE SYSTEM

Command Prompts and Getting Prompts Back

- What are command prompts?
 - A command prompt, also referred to simply as a prompt, is a short text at the start of the command line followed by prompt symbol on a command line interface



- To get your prompt back
 - **Ctrl + c**

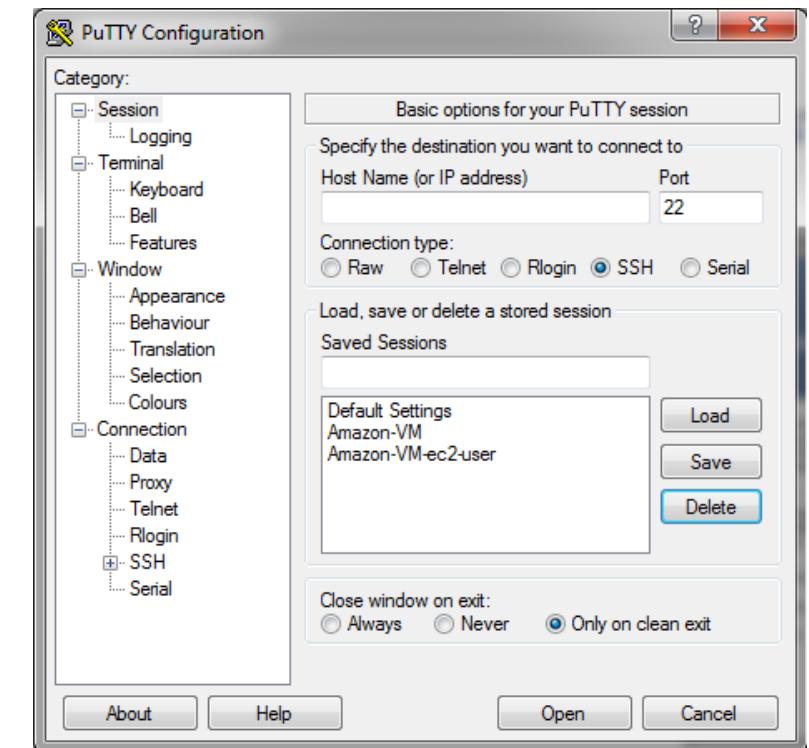
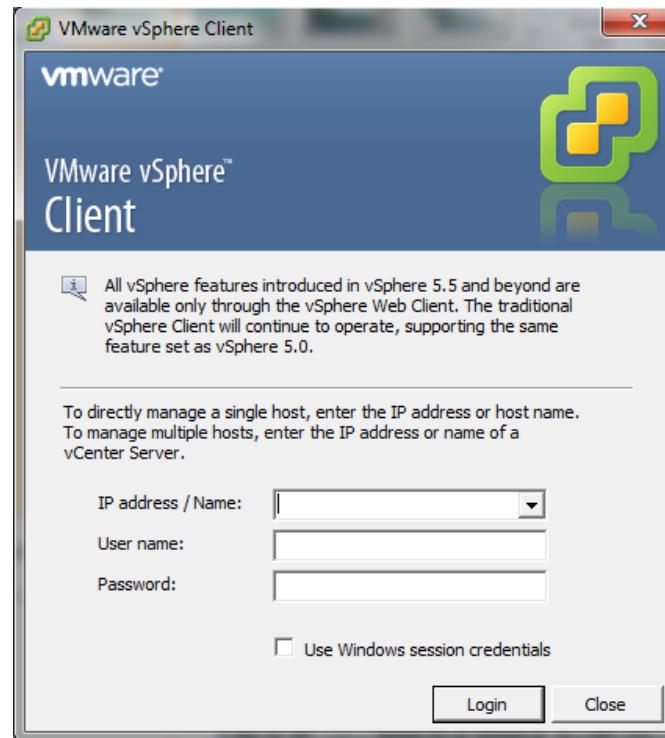
Access to Linux System

Each operating system has a different protocol or client that is used to access the system

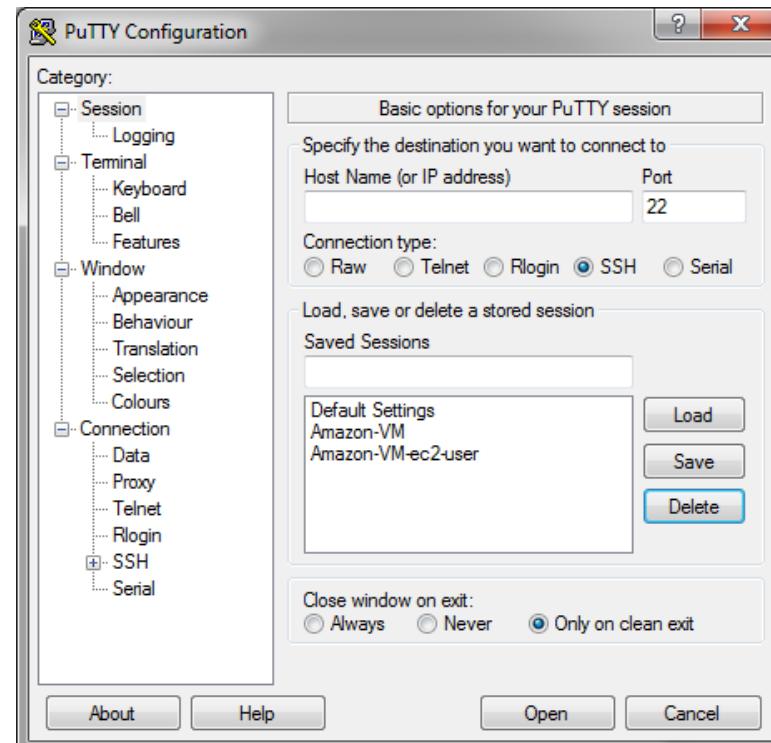
Example:

Windows	=	Remote Desktop (RDP)
VMware ESX	=	vSphere client
Linux	=	Putty, SecureCRT
		SSH from Linux to Linux

ACCESS TO LINUX SYSTEM



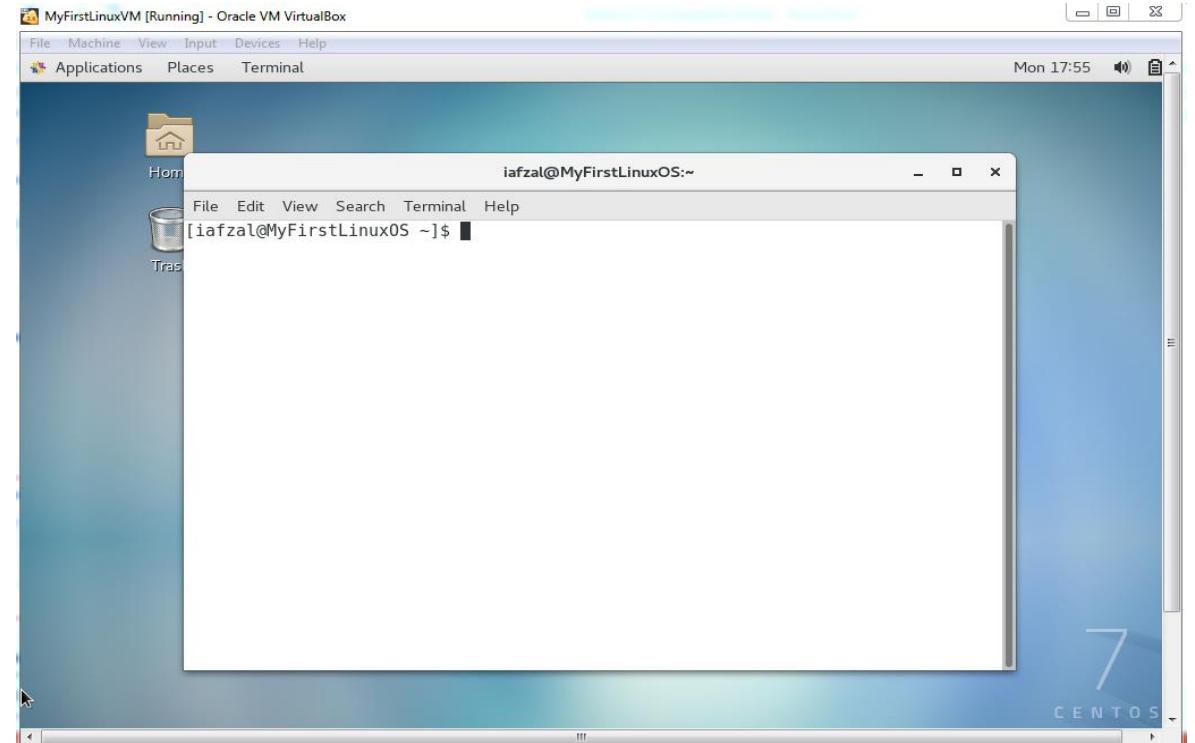
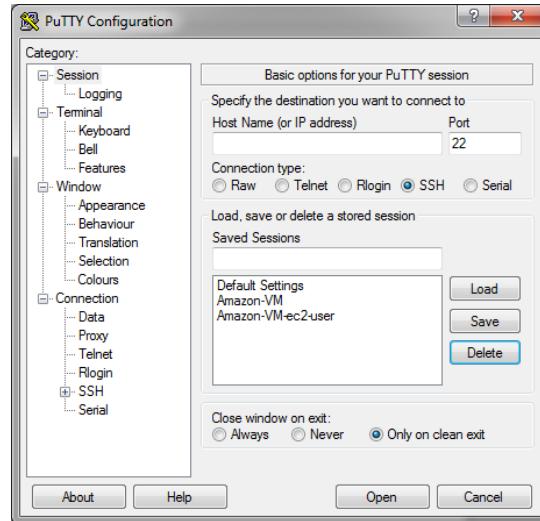
DOWNLOAD AND INSTALL PUTTY



Access to Linux from MAC

- Openup a terminal on your MAC
- Run the following command
 - **# ssh -l iafzal 192.168.56.101**

ACCESS TO LINUX VIA PUTTY



New Network Command (**ip**)

- CentOS/RHEL 5 or 6 = **ifconfig**
- CentOS/RHEL 7 = **ip**
- CentOS/RHEL 7.5 and up = **ifconfig** command has been deprecated
- To use ifconfig in 7.5 = “**yum install net-tools**”

Important Things to Remember in Linux

- Linux has super-user account called root
 - root is the most powerful account that can create, modify, delete accounts and make changes to system configuration files
- Linux is case-sensitive system
 - **ABC** is NOT same as **abc**
- Avoid using spaces when creating files and directories
- Linux kernel is not an operating system. It is a small software within Linux operating system that takes commands from users and pass them to system hardware or peripherals
- Linux is mostly CLI not GUI
- Linux is very flexible as compared to other operating systems.

Changing Password

- You should change your initial password as soon as you login

Command = **passwd userid**

Old password: - enter your current password

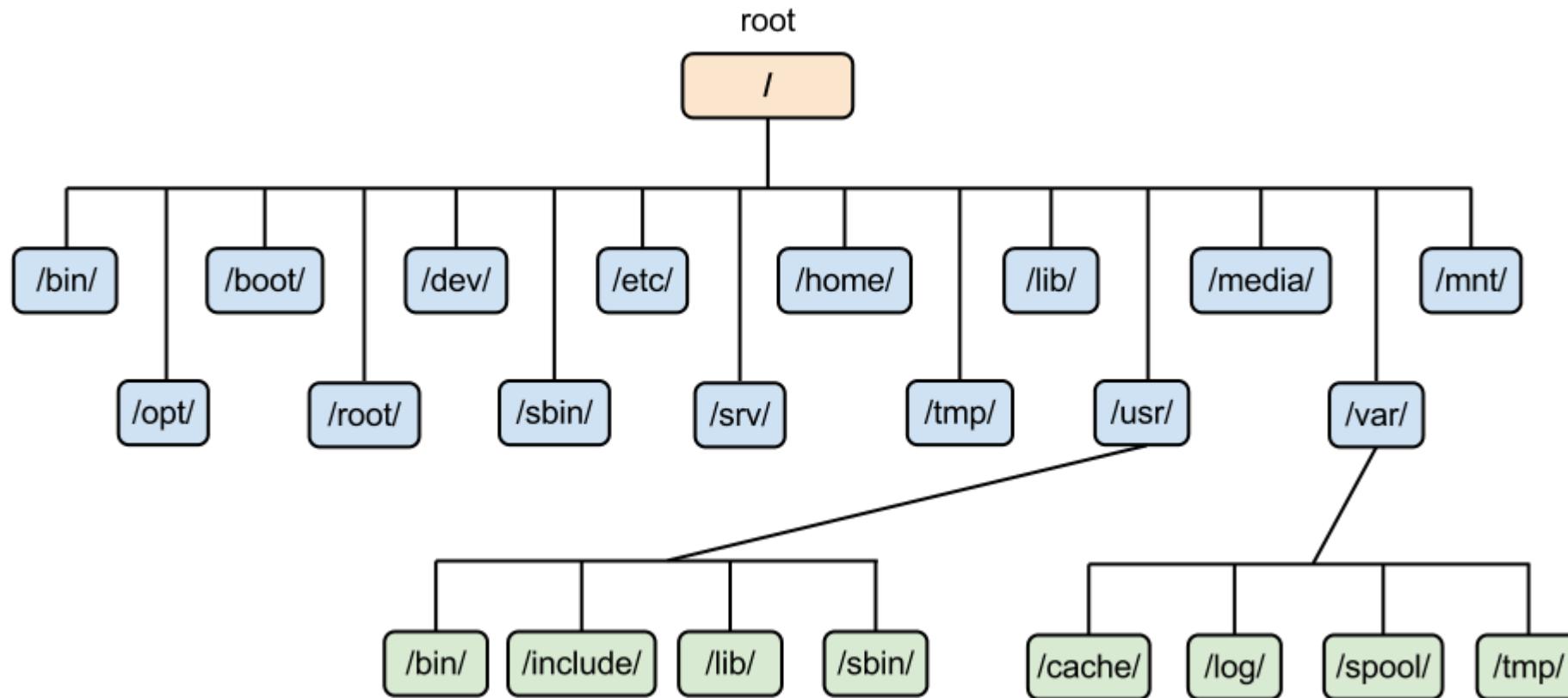
New password: - enter your new password

Retype new password: - re-enter your new password

Linux File System

- OS store data on disk drives using a structure called a filesystem, consisting of files, directories, and the information needed to access and locate them.
- There are many different types of filesystems. In general, improvements have been made to filesystems with new releases of operating systems, and each new filesystem has been given a different name:
E.g. ext3, ext4, XFS, NTFS, FAT etc.
- Linux filesystems store information in a hierarchy of directories and files

FILE SYSTEM STRUCTURE



File System Structure and its Description

/boot	Contains file that is used by the boot loader (grub.cfg)
/root	root user home directory. It is not same as /
/dev	System devices (e.g. disk, cdrom, speakers, flashdrive, keyboard etc.)
/etc	Configuration files
/bin → /usr/bin	Everyday user commands
/sbin → /usr/sbin	System/filesystem commands
/opt	Optional add-on applications (Not part of OS apps)
/proc	Running processes (Only exist in Memory)
/lib → usr/lib	C programming library files needed by commands and apps strace -e open pwd
/tmp	Directory for temporary files
/home	Directory for user
/var	System logs
/run	System daemons that start very early (e.g. systemd and udev) to store temporary runtime files like PID files
/mnt	To mount external filesystem. (e.g. NFS)
/media	For cdrom mounts.

Navigating File System

- When navigating a UNIX filesystem, there are a few important commands:

"cd"

"pwd"

"ls"

- "cd" stands for change directory. It is the primary command for moving you around the filesystem.
- "pwd" stands for print working directory. It tells you where you current location is.
- "ls" stands for list. It lists all the directories/files within a current working directory
- Using of TAB key to auto-complete

What is Root?

- There are 3 types of root on Linux system
 1. Root account: root is an account or a username on Linux machine and it is the most powerful account which has access to all commands and files
 2. Root as /: the very first directory in Linux is also referred as root directory
 3. Root home directory: the root user account also has a directory located in /root which is called root home directory

File System Paths

- There are two paths to navigate to a filesystem
 - ✓ Absolute Path
 - ✓ Relative Path
- An absolute path always begins with a "/". This indicates that the path starts at the root directory. An example of an absolute path is
`cd /var/log/httpd`

- A relative path does not begin with a "/". It identifies a location relative to your current position. An example of a relative path is:

```
cd /var
```

```
cd log
```

```
cd httpd
```

Directory Listing Attributes

Total columns = 9

Type	# of Links	Owner	Group	Size	Month	Day	Time	Name
drwxr-xr-x.	21	root	root	4096	Feb	27	13:33	var
lrwxrwxrwx.	1	root	root	7	Feb	27	13:15	bin
-rw-r--r--	1	root	root	0	Mar	2	11:15	testfile



The second column is the number of hard links to the file. For a directory, the number of hard links is the number of immediate subdirectories it has plus its parent directory and itself

Linux File Types

File Symbol	Meaning
-	Regular file
d	Directory
l	link
c	Special file or device file
s	socket
p	Named pipe
b	Block device

Creating Files and Directories

- Creating Files

- ✓ **touch**

- ✓ **cp**

- ✓ **vi**

- Creating Directories

- ✓ **mkdir**

Copying Directories

- Command to copy a directory
 - `cp`
- To copy a directory on Linux, you have to execute the “`cp`” command with the “`-R`” option for recursive and specify the source and destination directories to be copied
 - `cp -R <source_folder> <destination_folder>`

The “echo” command

- “echo” is one of the most commonly and widely used built-in command for Linux
- Just like the word echo, the command echo does the same thing
- “echo” command outputs the strings it is being passed as arguments
 - E.g. **echo hello world**
- It is also used to create add contents in a file using file redirects
 - E.g. **echo hello world > filename1.**

Find Files and Directories

- Two main commands are used to find files/directories
 - `find`
 - `locate`

Difference Between `find` and `locate`

- **locate** uses a prebuilt database, which should be regularly updated, while **find** iterates over a filesystem to locate files. Thus, locate is much faster than find , but can be inaccurate if the database (can be seen as a cache) is not updated
- To update locate database run **updatedb**

WildCards

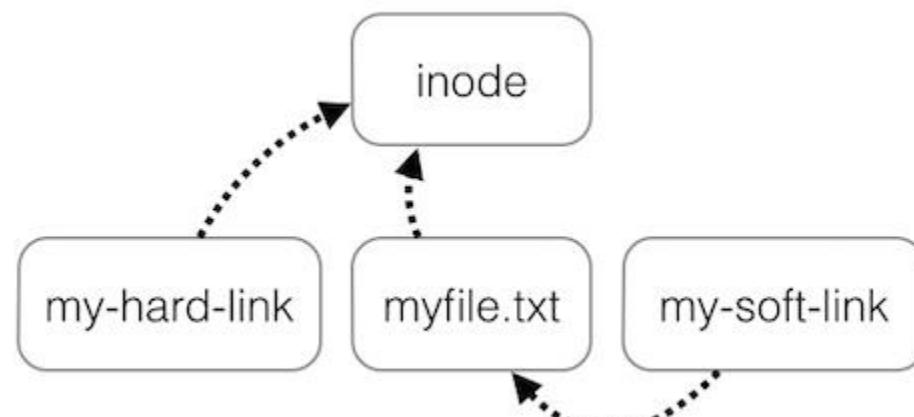
- A wildcard is a character that can be used as a substitute for any of a class of characters in a search

- * - represents zero or more characters
- ? - represents a single character
- [] - represents a range of characters

Soft and Hard Links

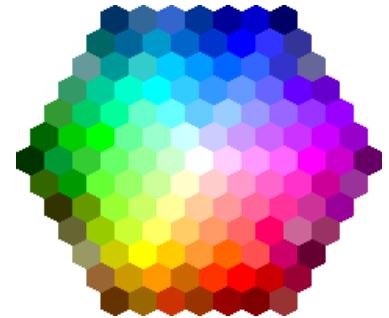
- inode = Pointer or number of a file on the hard disk
- Soft Link = Link will be removed if file is removed or renamed
- Hard Link = Deleting renaming or moving the original file will not affect the hard link

- ln
- ln -s



Filesystem Color Definition

```
iafzal@myfirstlinuxvm:~$ ls -ltr
total 296
drwxr-xr-x. 2 iafzal iafzal      6 Jan  8 21:44 Templates
drwxr-xr-x. 2 iafzal iafzal      6 Jan  8 21:44 Public
drwxr-xr-x. 2 iafzal iafzal      6 Jan  8 21:44 Downloads
drwxr-xr-x. 2 iafzal iafzal      6 Jan  8 21:44 Documents
drwxr-xr-x. 2 iafzal iafzal      6 Jan  8 21:44 Desktop
drwxr-xr-x. 2 iafzal iafzal      6 Jan  8 21:44 Videos
drwxr-xr-x. 2 iafzal iafzal      6 Jan  8 21:44 Pictures
drwxr-xr-x. 2 iafzal iafzal      6 Jan  8 21:44 Music
-rwxrwxr-x. 1 iafzal iafzal     47 Feb 11 19:10 homer
-rw----r--. 1 iafzal iafzal 247944 Feb 14 18:08 messages
drwxrwxr-x. 2 root  root       6 Feb 14 18:43 rootdir1
-rw-rw-r-- 1 iafzal iafzal    30 Feb 25 15:50 clients
-rwxrwxr-x 1 iafzal iafzal   164 Feb 25 16:00 checkclients
-rw-rw-r-- 1 iafzal iafzal      4 Feb 26 17:58 james
-rw-rw-r-- 1 iafzal iafzal  1608 Feb 26 19:16 peter
-rw-rw-r-- 1 iafzal iafzal  8649 Feb 26 19:19 ifconfig.txt
-rw-rw-r-- 1 iafzal iafzal  8719 Feb 26 19:23 ifconfig.file
-rw-rw-r-- 1 iafzal iafzal    186 Feb 27 19:42 seinfeld.bak
-rw-rw-r-- 1 iafzal iafzal    180 Feb 27 20:06 seinfeld
[iafzal@myfirstlinuxvm ~]$
```

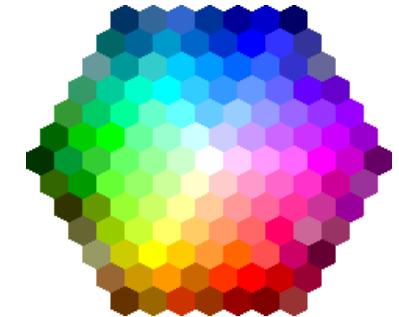


Filesystem Color Definition

- **Blue** = Directory
- **Green** = Executable or recognized data file
- **Sky Blue** = Symbolic link file



```
e.g. cd /home/iafzal  
touch ca  
cd /tmp  
ln -s /homeiafzal/ca
```



- Yellow with black background = Device

- Pink = Graphic image file
- Red = Archive file (tar)

- Red with black background = Broken Link



```
e.g. cd /home/iafzal  
touch ihulk  
ln -s ihulk /tmp/ihulk  
Absolute path for source file is missing
```

Linux Structure

Linux is a layered operating system. The innermost layer is the hardware that provides the services for the OS. The operating system, referred to in Linux as the kernel, interacts directly with the hardware and provides the services to the user programs. These user programs don't need to know anything about the hardware. They just need to know how to interact with the kernel and it's up to the kernel to provide the desired service. One of the big appeals of Linux to programmers has been that most well written user programs are independent of the underlying hardware, making them readily portable to new systems.

User programs interact with the kernel through a set of standard system calls. These system calls request services to be provided by the kernel. Such services would include accessing a file: open close, read, write, link, or execute a file; starting or updating accounting records; changing ownership of a file or directory; changing to a new directory; creating, suspending, or killing a process; enabling access to hardware devices; and setting limits on system resources.

Linux is a multi-user, multi-tasking operating system. You can have many users logged into a system simultaneously, each running many programs. It's the kernel's job to keep each process and user separate and to regulate access to system hardware, including cpu, memory, disk and other I/O devices.

Linux vs. Windows

Linux and Windows. Each has its own set of unique features, advantages and disadvantages. While it is difficult to say which one is the better choice, it is not as difficult to answer which is the better choice given your needs.

Note: The operating system that you use on your desktop computer (the vast majority of people use some flavor of Windows) has absolutely nothing to do with the one that your host needs to serve your web site. Most personal sites are created with MS FrontPage and even although that is a Microsoft product, it can be hosted perfectly on a LINUX web server with FrontPage Extensions installed.

Stability:

LINUX systems (we actually use Linux but for comparison purposes they are identical) are hands-down the winner in this category. There are many factors here but to name just a couple big ones: in our experience LINUX handles high server loads better than Windows and LINUX machines seldom require reboots while Windows is constantly needing them. Servers running on LINUX enjoy extremely high up-time and high availability/reliability.

Performance:

While there is some debate about which operating system performs better, in our experience both perform comparably in low-stress conditions however LINUX servers under high load (which is what is important) are superior to Windows.

Scalability:

Web sites usually change over time. They start off small and grow as the needs of the person or organization running them grow. While both platforms can often adapt to your growing needs, Windows hosting is more easily made compatible with LINUX-based programming features like PHP and MySQL. LINUX-based web software is not always 100% compatible with Microsoft technologies like .NET and VB development. Therefore if you wish to use these, you should choose Windows web hosting.

Compatibility:

Web sites designed and programmed to be served under a LINUX-based web server can easily be hosted on a Windows server, whereas the reverse is not always true. This makes programming for LINUX the better choice.

Price:

Servers hosting your web site require operating systems and licenses just like everyone else. Windows 2003 and other related applications like SQL Server each cost a significant amount of money; on the other hand, Linux is a free operating system to download, install and operate. Windows hosting results in being a more expensive platform.

Conclusion:

To sum it up, LINUX-based hosting is more stable, performs faster and more compatible than Windows-based hosting. You only need Windows hosting if you are going to developing in .NET or Visual Basic, or some other application that limits your choices

Logging On To System

- Before you can begin to use the system you will need to have a valid username and a password. Assignment of usernames and initial passwords is typically handled by the System Administrator
- Your username, also called a userid, should be unique and should not change. Initial passwords can be anything and should be changed after your first login.

To login to your account

- Type your username at the login prompt, initial of your first name followed by last name (e.g iafzal). LINUX is case sensitive - if your username is kellyk do not type KellyK . Press the RETURN or ENTER key after typing your username.
- When the password prompt appears, type in your password. Your password is never displayed on the screen as a security measure. It also is case sensitive. Press the RETURN or ENTER key after entering your password.
- What happens after you successfully login depends upon your system, many LINUX systems will display a login banner or "message of the day". Make a habit of reading this since it may contain important information about the system.
- Other LINUX systems will automatically configure your environment and open one or more windows for you to do work in.
- You should see a prompt - usually a percent sign (%) or dollar sign (\$). This is called the "shell prompt" (the shell is discussed in detail later). It indicates that the system is ready to accept commands from you.

If your login attempt was unsuccessful, there are several possible reasons:

- You made a typing error while entering your username or password
- The CAPS LOCK key is on and everything is being sent to the system in uppercase letters.
- You have an expired or invalid username or password, or the system security has changed
- There are system problems

Example of user login

```
login: kellyk
kellyk's Password:
*****
* Welcome to the Linux Systems Training Class
*****
*
```

```
* Hello! (Greetings)
*
* System maintenance is scheduled today from 2:00
* until 4:00 pm EST
*
*           (Thank you very much)
*
*****
```

Your Home Directory

- Each user has a unique "home" directory. Your home directory is that part of the file system reserved for your files.
- After login, you are "put" into your home directory automatically. This is where you start your work.
- You are in control of your home directory and the files which reside there. You are also in control of the file access permissions (discussed later) to the files in your home directory. Generally, you alone should be able to create/delete/modify files in your home directory. Others may have permission to read or execute your files as you determine.
- In most LINUX systems, you can "move around" or navigate to other parts of the file system outside of your home directory. This depends upon how the file permissions have been set by others and/or the System Administrator

Linux File System

A file system is a logical collection of files on a partition or disk. A partition is a container for information and can span an entire hard drive if desired.

Your hard drive can have various partitions which usually contains only one file system, such as one file system housing the / file system or another containing the /home file system.

One file system per partition allows for the logical maintenance and management of differing file systems.

Everything in Linux is considered to be a file, including physical devices such as DVD-ROMs, USB devices, floppy drives, and so forth.

Directory Structure:

Linux uses a hierarchical file system structure, much like an upside-down tree, with root (/) at the base of the file system and all other directories spreading from there.

A LINUX filesystem is a collection of files and directories that has the following properties:

It has a root directory (/) that contains other files and directories.

Each file or directory is uniquely identified by its name, the directory in which it resides, and a unique identifier, typically called an inode.

By convention, the root directory has an inode number of 2 and the lost+found directory has an inode number of 3. Inode numbers 0 and 1 are not used. File inode numbers can be seen by specifying the -i option to ls command.

It is self contained. There are no dependencies between one filesystem and any other.

File System:

What are filesystems?

A filesystem is the methods and data structures that an operating system uses to keep track of files on a disk or partition; that is, the way the files are organized on the disk. The word is also used to refer to a partition or disk that is used to store the files or the type of the filesystem. Thus, one might say "I have two filesystems" meaning one has two partitions on which one stores files, or that one is using the "extended filesystem", meaning the type of the filesystem.

The difference between a disk or partition and the filesystem it contains is important. A few programs (including, reasonably enough, programs that create filesystems) operate directly on the raw sectors of a disk or partition; if there is an existing file system there it will be destroyed or seriously corrupted. Most programs operate on a filesystem, and therefore won't work on a partition that doesn't contain one (or that contains one of the wrong types).

Before a partition or disk can be used as a filesystem, it needs to be initialized, and the bookkeeping data structures need to be written to the disk. This process is called making a filesystem.

Most LINUX filesystem types have a similar general structure, although the exact details vary quite a bit. The central concepts are superblock, inode , data block, directory block , and indirection block. The superblock contains information about the filesystem as a whole, such as its size (the exact information here depends on the filesystem). An inode contains all information about a file, except its name. The name is stored in the directory, together with the number of the inode. A directory entry consists of a filename and the number of the inode which represents the file. The inode contains the numbers of several data blocks, which are used to store the data in the file. There is space only for a few data block numbers in the inode, however, and if more are needed, more space for pointers to the data blocks is allocated dynamically. These dynamically allocated blocks are indirect blocks; the name indicates that in order to find the data block, one has to find its number in the indirect block first.

LINUX filesystems usually allow one to create a hole in a file (this is done with the lseek() system call; check the manual page), which means that the filesystem just pretends that at a particular place in the file there is just zero bytes, but no actual disk sectors are reserved for that place in the file (this means that the file will use a bit less disk space). This happens especially often for small binaries, Linux shared libraries, some databases, and a few other special cases. (Holes are implemented by storing a special value as the address of the data block in the indirect block or inode. This special address means that no data block is allocated for that part of the file, ergo, there is a hole in the file.)

Comparing Filesystem Features

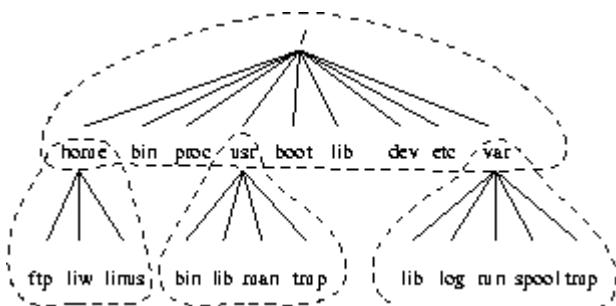
FS Name	Year Introduced	Original OS	Max File Size	Max FS Size	Journaling
FAT16	1983	MSDOS V2	4GB	16MB to 8GB	N
FAT32	1997	Windows 95	4GB	8GB to 2TB	N

FS Name	Year Introduced	Original OS	Max File Size	Max FS Size	Journaling
HPFS	1988	OS/2	4GB	2TB	N
NTFS	1993	Windows NT	16EB	16EB	Y
HFS+	1998	Mac OS	8EB	?	N
UFS2	2002	FreeBSD	512GB to 32PB	1YB	N
ext2	1993	Linux	16GB to 2TB4	2TB to 32TB	N
ext3	1999	Linux	16GB to 2TB4	2TB to 32TB	Y
ReiserFS3	2001	Linux	8TB8	16TB	Y
ReiserFS4	2005	Linux	?	?	Y
XFS	1994	IRIX	9EB	9EB	Y
JFS	?	AIX	8EB	512TB to 4PB	Y
VxFS	1991	SVR4.0	16EB	?	Y
ZFS	2004	Solaris 10	1YB	16EB	N

This topic is loosely based on the *Filesystems Hierarchy Standard* (FHS), which attempts to set a standard for how the directory tree in a Linux system should be organized. Such a standard has the advantage that it will be easier to write or port software for Linux, and to administer Linux machines, since everything should be in standardized places. There is no authority behind the standard that forces anyone to comply with it, but it has gained the support of many Linux distributions. It is not a good idea to break with the FHS without very compelling reasons. The FHS attempts to follow Linux tradition and current trends, making Linux systems familiar to those with experience with other Linux systems, and vice versa.

The full directory tree is intended to be breakable into smaller parts, each capable of being on its own disk or partition, to accommodate to disk size limits and to ease backup and other system administration tasks. The major parts are the root (/), /usr , /var , and /home filesystems (*see the following figure*). Each part has a different purpose. The directory tree has been designed so that it works well in a network of Linux machines which may share some parts of the filesystems over a read-only device (e.g., a CD-ROM), or over the network with NFS.

Parts of a Linux directory tree. Dashed lines indicate partition limits



The roles of the different parts of the directory tree are described below

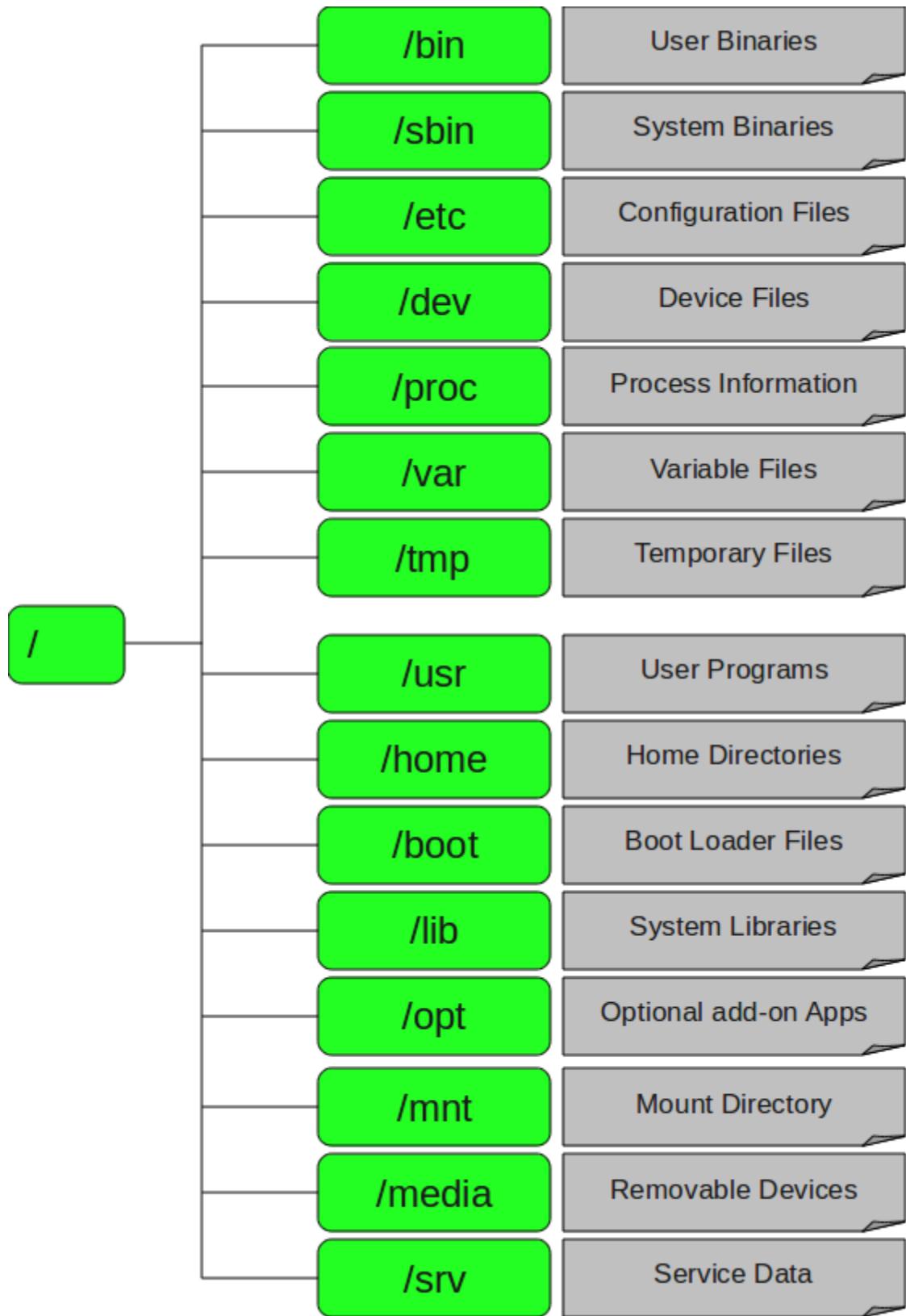
- The root filesystem is specific for each machine (it is generally stored on a local disk, although it could be a ramdisk or network drive as well) and contains the files that are necessary for booting the system up, and to bring it up to such a state that the other filesystems may be mounted. The contents of the root filesystem will therefore be sufficient for the single user state. It will also contain tools for fixing a broken system, and for recovering lost files from backups.
- The /usr filesystem contains all commands, libraries, manual pages, and other unchanging files needed during normal operation. No files in /usr should be specific for any given machine, nor should they be modified during normal use. This allows the files to be shared over the network, which can be cost-effective since it saves disk space (there can easily be hundreds of megabytes, increasingly multiple gigabytes in /usr). It can make administration easier (only the master /usr needs to be changed when updating an application, not each machine separately) to have /usr network mounted. Even if the filesystem is on a local disk, it could be mounted read-only, to lessen the chance of filesystem corruption during a crash.
- The /var filesystem contains files that change, such as spool directories (for mail, news, printers, etc), log files, formatted manual pages, and temporary files. Traditionally everything in /var has been somewhere below /usr , but that made it impossible to mount /usr read-only.
- The /home filesystem contains the users' home directories, i.e., all the real data on the system. Separating home directories to their own directory tree or filesystem makes backups easier; the other parts often do not have to be backed up, or at least not as often as they seldom change. A big /home might have to be broken across several filesystems, which requires adding an extra naming level below /home, for example /home/students and /home/staff.

Although the different parts have been called filesystems above, there is no requirement that they actually be on separate filesystems. They could easily be kept in a single one if the system is a small single-user system and the user wants to keep things simple. The directory tree might also be divided into filesystems differently, depending on how large the disks are, and how space is allocated for various purposes. The important part, though, is that all the standard *names* work; even if, say, /var and /usr are actually on the same partition, the names /usr/lib/libc.a and /var/log/messages must work, for example by moving files below /var into /usr/var, and making /var a symlink to /usr/var.

The Linux filesystem structure groups files according to purpose, i.e., all commands are in one place, all data files in another, documentation in a third, and so on. An alternative would be to group files according to the program they belong to, i.e., all Emacs files would be in one directory, all TeX in another, and so on. The problem with the latter approach is that it makes it difficult to share files (the program directory often contains both static and sharable and changing and non-sharable files), and sometimes to even find the files (e.g., manual pages in a huge number of places, and making the manual page programs find all of them is a maintenance nightmare).

The root filesystem should generally be small, since it contains very critical files and a small, infrequently modified filesystem has a better chance of not getting corrupted. A corrupted root filesystem will generally mean that the system becomes unbootable except with special measures (e.g., from a floppy), so you don't want to risk it.

The root directory generally doesn't contain any files, except perhaps on older systems where the standard boot image for the system, usually called /vmlinuz was kept there. (Most distributions have moved those files the the /boot directory.



1. / – Root

- Every single file and directory starts from the root directory.
- Only root user has write privilege under this directory.
- Please note that /root is root user's home directory, which is not same as /.

2. /bin – User Binaries

- Contains binary executables.
- Common linux commands you need to use in single-user modes are located under this directory.
- Commands used by all the users of the system are located here.
- For example: ps, ls, ping, grep, cp.

3. /sbin – System Binaries

- Just like /bin, /sbin also contains binary executables.
- But, the linux commands located under this directory are used typically by system administrator, for system maintenance purpose.
- For example: iptables, reboot, fdisk, ifconfig, swapon

4. /etc – Configuration Files

- Contains configuration files required by all programs.
- This also contains startup and shutdown shell scripts used to start/stop individual programs.
- For example: /etc/resolv.conf, /etc/logrotate.conf

5. /dev – Device Files

- Contains device files.
- These include terminal devices, usb, or any device attached to the system.
- For example: /dev/tty1, /dev/usbmon0

6. /proc – Process Information

- Contains information about system process.
- This is a pseudo filesystem contains information about running process. For example: /proc/{pid} directory contains information about the process with that particular pid.
- This is a virtual filesystem with text information about system resources. For example: /proc/uptime

7. /var – Variable Files

- var stands for variable files.

- Content of the files that are expected to grow can be found under this directory.
- This includes — system log files (/var/log); packages and database files (/var/lib); emails (/var/mail); print queues (/var/spool); lock files (/var/lock); temp files needed across reboots (/var/tmp);

8. /tmp – Temporary Files

- Directory that contains temporary files created by system and users.
- Files under this directory are deleted when system is rebooted.

9. /usr – User Programs

- Contains binaries, libraries, documentation, and source-code for second level programs.
- /usr/bin contains binary files for user programs. If you can't find a user binary under /bin, look under /usr/bin. For example: at, awk, cc, less, scp
- /usr/sbin contains binary files for system administrators. If you can't find a system binary under /sbin, look under /usr/sbin. For example: atd, cron, sshd, useradd, userdel
- /usr/lib contains libraries for /usr/bin and /usr/sbin
- /usr/local contains users programs that you install from source. For example, when you install apache from source, it goes under /usr/local/apache2

10. /home – Home Directories

- Home directories for all users to store their personal files.
- For example: /home/john, /home/nikita

11. /boot – Boot Loader Files

- Contains boot loader related files.
- Kernel initrd, vmlinuz, grub files are located under /boot
- For example: initrd.img-2.6.32-24-generic, vmlinuz-2.6.32-24-generic

12. /lib – System Libraries

- Contains library files that supports the binaries located under /bin and /sbin
- Library filenames are either ld* or lib*.so.*
- For example: ld-2.11.1.so, libncurses.so.5.7

13. /opt – Optional add-on Applications

- opt stands for optional.
- Contains add-on applications from individual vendors.
- add-on applications should be installed under either /opt/ or /opt/ sub-directory.

14. /mnt – Mount Directory

- Temporary mount directory where sysadmins can mount filesystems.

15. /media – Removable Media Devices

- Temporary mount directory for removable devices.
- For examples, /media/cdrom for CD-ROM; /media/floppy for floppy drives; /media/cdrecorder for CD writer

16. /srv – Service Data

- srv stands for service.
- Contains server specific services related data.
- For example, /srv/cvs contains CVS related data

File Names

- LINUX permits file names to use most characters, but avoid spaces, tabs and characters that have a special meaning to the shell, such as:

& ; () | ? \ ' " ` [] { } < > \$ - ! /

- Case Sensitivity: uppercase and lowercase are not the same! These are three different files:

NOVEMBER November november

- Length: can be up to 256 characters

- Extensions: may be used to identify types of files

libc.a	- archive, library file
program.c	- C language source file
alpha2.f	- Fortran source file
xwd2ps.o	- Object/executable code
mygames.Z	- Compressed file

- Hidden Files: have names that begin with a dot (.) For example:

.cshrc .login .mailrc .mwmrc

- Uniqueness: as children in a family, no two files with the same parent directory can have the same name. Files located in separate directories can have identical names.

- Reserved Filenames:

/	- the root directory (slash)
.	- current directory (period)
..	- parent directory (double period)
~	- your home directory (tilde)

Passwords Standards

When your account is issued, you will be given an initial password. It is important for system and personal security that the password for your account be changed to something of your choosing. The command for changing a password is "passwd". You will be asked both for your old password and to type your new selected password twice. If you mistype your old password or do not type your new password the same way twice, the system will indicate that the password has not been changed. Some system administrators have installed programs that check for appropriateness of password (is it cryptic enough for reasonable system security). A password change may be rejected by this program. When choosing a password, it is important that it be something that could not be guessed -- either by somebody unknown to you trying to break in, or by an acquaintance who knows you. Suggestions for choosing and using a password follow:

Don't

- use a word (or words) in any language
- use a proper name
- use information that can be found in your wallet
- use information commonly known about you (car license, pet name, etc)
- use control characters. Some systems can't handle them
- write your password anywhere
- ever give your password to *anybody*

Do

- use a mixture of character types (alphabetic, numeric, special)
- use a mixture of upper case and lower case
- use at least 6 characters
- choose a password you can remember
- change your password often
- make sure nobody is looking over your shoulder when you are entering your password

Change Password in LINUX

How do I change the password in LINUX?

To modify a user's password or your own password in LINUX use the passwd command. Open the terminal and then type the passwd command entering the new password, the characters entered do not display on screen, in order to avoid the password being seen by a passer-by. The passwd command prompts for the new password twice in order to detect any typing errors. The encrypted password is stored in /etc/shadow file.

Change Any Users Password

Login as the root user and type the command:

```
# passwd userName  
# passwd vivek  
# passwd tom
```

Sample outputs:

```
Enter new LINUX password:  
Retype new LINUX password:  
passwd: password updated successfully
```

Change Your Own Password

Simply type the passwd command:

```
$ passwd
```

Sample outputs:

```
(current) LINUX password:  
Enter new LINUX password:  
Retype new LINUX password:  
passwd: password updated successfully
```

Difference between locate and find command in Linux

Two popular commands for locating files on Linux are find and locate. Depending on the size of your file system and the depth of your search, the find command can sometime take a long time to scan all of the data. For example, if you search your entire filesystem for the files named data.txt:

```
# find / -name data.txt
```

More likely than not, this will take on the order of minutes, if not longer to return. A quicker method is to use the locate command:

```
# locate data.txt
```

However, this efficiency comes at a cost, the data reported in the output of locate isn't as fresh as the data reported by the find command. By default, the system will run updatedb which takes a snapshot of the system files once a day, locate uses this snapshot to quickly report what files are where. However, recent file additions or removals (within 24 hours) are not recorded in the snapshot and are unknown to locate.

The find command has a number of options and is very configurable. There are many ways to reduce the depth and breadth of your search and make it more efficient.

locate uses a previously built database. If database is not updated then locate command will not show the output. to sync the database it is must to execute updatedb command.

```
# updatedb
```

How to Use Wildcards

A wildcard is a character that can be used as a substitute for any of a class of characters in a search, thereby greatly increasing the flexibility and efficiency of searches.

Wildcards are commonly used in shell commands in Linux and other Unix-like operating systems. A shell is a program that provides a text-only user interface and whose main function is to execute commands typed in by users and display their results.

Wildcards are also used in regular expressions and programming languages. Regular expressions are a pattern matching system that uses strings (i.e., sequences of characters) constructed according to pre-defined syntax rules to find desired strings in text.

The term wildcard or wild card was originally used in card games to describe a card that can be assigned any value that its holder desires. However, its usage has spread so that it is now used to describe an unknown or unpredictable factor in a variety of fields.

Star Wildcard

Three types of wildcards are used with Linux commands. The most frequently employed and usually the most useful is the star wildcard, which is the same as an asterisk (*). The star wildcard has the broadest meaning of any of the wildcards, as it can represent zero characters, all single characters or any string.

As an example, the file command provides information about any filesystem object (i.e., file, directory or link) that is provided to it as an argument (i.e., input). Because the star wildcard represents every string, it can be used as the argument for file to return information about every object in the specified directory. Thus, the following would display information about every object in the current directory (i.e., the directory in which the user is currently working):

```
file *
```

If there are no matches, an error message is returned, such as *: can't stat '*' (No such file or directory).. In the case of this example, the only way that there would be no matches is if the directory were empty.

Wildcards can be combined with other characters to represent parts of strings. For example, to represent any filesystem object that has a .jpg filename extension, *.jpg would be used. Likewise, a* would represent all objects that begin with a lower case (i.e., small) letter a.

As another example, the following would tell the ls command (which is used to list files) to provide the names of all files in the current directory that have an .html or a .txt extension:

```
ls *.html *.txt
```

Likewise, the following would tell the rm command (which is used to remove files and directories) to delete all files in the current directory that have the string xxx in their name:

```
rm *xxx*
```

Question Mark Wildcard

The question mark (?) is used as a wildcard character in shell commands to represent exactly one character, which can be any single character. Thus, two question marks in succession would represent any two characters in succession, and three question marks in succession would represent any string consisting of three characters.

Thus, for example, the following would return data on all objects in the current directory whose names, inclusive of any extensions, are exactly three characters in length:

```
file ???
```

And the following would provide data on all objects whose names are one, two or three characters in length:

```
file ? ?? ???
```

As is the case with the star wildcard, the question mark wildcard can be used in combination with other characters. For example, the following would provide information about all objects in the current directory that begin with the letter a and are five characters in length:

```
file a????
```

The question mark wildcard can also be used in combination with other wildcards when separated by some other character. For example, the following would return a list of all files in the current directory that have a three-character filename extension:

```
ls *.???
```

Square Brackets Wildcard

The third type of wildcard in shell commands is a pair of square brackets, which can represent any of the characters enclosed in the brackets. Thus, for example, the following would provide information about all objects in the current directory that have an x, y and/or z in them:

```
file *[xyz]*
```

And the following would list all files that had an extension that begins with x, y or z:

```
ls *.[xyz]*
```

The same results can be achieved by merely using the star and question mark wildcards. However, it is clearly more efficient to use the bracket wildcard.

When a hyphen is used between two characters in the square brackets wildcard, it indicates a range inclusive of those two characters. For example, the following would provide information about all of the objects in the current directory that begin with any letter from a through f:

```
file [a-f]*
```

And the following would provide information about every object in the current directory whose name includes at least one numeral:

```
file *[0-9]*
```

The use of the square brackets to indicate a range can be combined with its use to indicate a list. Thus, for example, the following would provide information about all filesystem objects whose names begin with any letter from a through c or begin with s or t:

```
file [a-cst]*
```

Likewise, multiple sets of ranges can be specified. Thus, for instance, the following would return information about all objects whose names begin with the first three or the final three lower case letters of the alphabet:

```
file [a-cx-z]*
```

Sometimes it can be useful to have a succession of square bracket wildcards. For example, the following would display all filenames in the current directory that consist of jones followed by a three-digit number:

```
ls jones[0-9][0-9][0-9]
```

Other Wild Cards

\ (backslash) = is used as an "escape" character, i.e. to protect a subsequent special character. Thus, "\\\" searches for a backslash. Note you may need to use quotation marks and backslash(es).

^ (caret) = means "the beginning of the line". So "^a" means find a line starting with an "a".

\$ (dollar sign) = means "the end of the line". So "a\$" means find a line ending with an "a".

For example, this command searches the file myfile for lines starting with an "s" and ending with an "n", and prints them to the standard output (screen):

```
cat myfile | grep '^s.*n$'
```

Soft Link and Hard Links

Example:

Create two files:

```
$ touch blah1  
$ touch blah2
```

Enter some data into them:

```
$ echo "Cat" > blah1  
$ echo "Dog" > blah2
```

And as expected:

```
$ cat blah1; cat blah2  
Cat  
Dog
```

Let's create hard and soft links:

```
$ ln blah1 blah1-hard  
$ ln -s blah2 blah2-soft
```

Let's see what just happened:

```
$ ls -l  
blah1  
blah1-hard  
blah2  
blah2-soft -> blah2
```

Changing the name of blah1 does not matter:

```
$ mv blah1 blah1-new  
$ cat blah1-hard  
Cat
```

blah1-hard points to the inode, the contents, of the file - that wasn't changed.

```
$ mv blah2 blah2-new  
$ ls blah2-soft  
blah2-soft  
$ cat blah2-soft
```

```
cat: blah2-soft: No such file or directory
```

The contents of the file could not be found because the soft link points to the name, that was changed, and not to the contents.

Similarly, If blah1 is deleted, blah1-hard still holds the contents; if blah2 is deleted, blah2-soft is just a link to a non-existing file.

List folders and files in a directory

Written By: Alexandros Mavridis

Contents

Listing Folders

[Non Hidden Folders](#)

The command:

ls - list directory contents

Information Commands:

ls --version
ls --help
info ls
man ls

[Hidden Folders](#)

Page 1

[Non Hidden And Hidden Folders](#)

Page 3

Page 4

Listing Files

[Non Hidden Files](#)

Page 5

[Hidden Files](#)

Page 7

[Non Hidden And Hidden Files](#)

Page 8

Listing Folders and Files

[Non Hidden Folders and Files](#)

Page 9

[Hidden Folders And Files](#)

Page 13

[Non Hidden And Hidden Folders And Files](#)

Page 20

[Sources](#)

Page 23

Options Used In This Document

-r, --reverse

reverse order while sorting

-l use a long listing format

-t sort by modification time, newest first

-i, --inode

print the index number of each file

-a, --all

do not ignore entries starting with .

-d, --directory

list directories themselves, not their contents

-p, --indicator-style=slash

append / indicator to directories

--group-directories-first

group directories before files

All the commands in the current document can be | to wc -l command for printing number of folders or files, instead of folders and files themselves. For example:

ls -d */ | wc -l

A. Listing Folders

Non hidden folders

Command	Output
ls -d */	Prints all non hidden folders in the current working directory in alphabetical order.
ls -dr */	Prints all non hidden folders in the current working directory in reverse alphabetical order.

ls -dl */	
ls -l grep ^d	Prints in detail all non hidden folders in the current working directory in alphabetical order.
ls -l awk '{if (\$1 ~ /d/) print \$0}'	
ls -dlr */	
ls -lr grep ^d	Prints in detail all non hidden folders in the current working directory in reverse alphabetical order.
ls -lr awk '{if (\$1 ~ /d/) print \$0}'	
ls -dt */	Prints all non hidden folders in the current working directory in chronological order, going from newest to oldest.
ls -dtr */	Prints all non hidden folders in the current working directory in reverse chronological order, going from oldest to newest.
ls -dlt */	
ls -lt grep ^d	Prints in detail all non hidden folders in the current working directory in chronological order, going from newest to oldest.
ls -lt awk '{if (\$1 ~ /d/) print \$0}'	
ls -dltr */	
ls -ltr grep ^d	Prints in detail all non hidden folders in the current working directory in reverse chronological order, going from oldest to newest.
ls -ltr awk '{if (\$1 ~ /d/) print \$0}'	
ls -di */	Prints all non hidden folders in the current working directory, including inode numbers, in alphabetical order.
ls -dri */	Prints all non hidden folders in the current working directory, including inode numbers, in reverse alphabetical order.
ls -dli */	Prints in detail all non hidden folders in the current working directory, including inode numbers, in alphabetical order.
ls -drli */	Prints in detail all non hidden folders in the current working directory, including inode numbers, in reverse alphabetical order.
ls -dti */	Prints all non hidden folders in the current working directory, including inode numbers, in chronological order, going from newest to oldest.
ls -dtri */	Prints all non hidden folders in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest.
ls -dlti */	Prints in detail all non hidden folders in the current working directory, including inode numbers, in chronological order, going from newest to oldest.
ls -dltri */	Prints in detail all non hidden folders in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest.

Hidden folders

Command	Output
ls -d .*/	Prints all hidden folders in the current working directory in alphabetical order.
ls -dr .*/	Prints all hidden folders in the current working directory in reverse alphabetical order.
ls -dl .*/	Prints in detail all hidden folders in the current working directory in alphabetical order.
ls -dlr .*/	Prints in detail all hidden folders in the current working directory in reverse alphabetical order.
ls -dt .*/	Prints all hidden folders in the current working directory in chronological order, going from newest to oldest.
ls -dtr .*/	Prints all hidden folders in the current working directory in reverse chronological order, going from oldest to newest.
ls -dlt .*/	Prints in detail all hidden folders in the current working directory in chronological order, going from newest to oldest.
ls -dltr .*/	Prints in detail all hidden folders in the current working directory in reverse chronological order, going from oldest to newest.
ls -di .*/	Prints all hidden folders in the current working directory, including inode numbers, in alphabetical order.
ls -dri .*/	Prints all hidden folders in the current working directory, including inode numbers, in reverse alphabetical order.
ls -dli .*/	Prints in detail all hidden folders in the current working directory, including inode numbers, in alphabetical order.
ls -drl .*/	Prints in detail all hidden folders in the current working directory, including inode numbers, in reverse alphabetical order.
ls -dti .*/	Prints all hidden folders in the current working directory, including inode numbers, in chronological order, going from newest to oldest.
ls -dri .*/	Prints all hidden folders in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest.

<code>ls -dlti */</code>	Prints in detail all hidden folders in the current working directory, including inode numbers, in chronological order, going from newest to oldest.
<code>ls -dltri */</code>	Prints in detail all hidden folders in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest.

Non hidden and hidden folders

Command	Output
<code>ls -d */ .*/</code>	Prints all non hidden and hidden folders in the current working directory in alphabetical order.
<code>ls -dr */ .*/</code>	Prints all non hidden and hidden folders in the current working directory in reverse alphabetical order.
<code>ls -dl */ .*/</code>	Prints in detail all non hidden and hidden folders in the current working directory in alphabetical order.
<code>ls -dlr */ .*/</code>	Prints in detail all non hidden and hidden folders in the current working directory in reverse alphabetical order.
<code>ls -dt */ .*/</code>	Prints all non hidden and hidden folders in the current working directory in chronological order, going from newest to oldest.
<code>ls -dtr */ .*/</code>	Prints all non hidden and hidden folders in the current working directory in reverse chronological order, going from oldest to newest.
<code>ls -dlt */ .*/</code>	Prints in detail all non hidden and hidden folders in the current working directory in chronological order, going from newest to oldest.
<code>ls -dltr */ .*/</code>	Prints in detail all non hidden and hidden folders in the current working directory in reverse chronological order, going from oldest to newest.
<code>ls -di */ .*/</code>	Prints all non hidden and hidden folders in the current working directory, including inode numbers, in alphabetical order.

ls -dri */ .*/	Prints all non hidden and hidden folders in the current working directory, including inode numbers, in reverse alphabetical order.
ls -dli */ .*/	Prints in detail all non hidden and hidden folders in the current working directory, including inode numbers, in alphabetical order.
ls -drli */ .*/	Prints in detail all non hidden and hidden folders in the current working directory, including inode numbers, in reverse alphabetical order.
ls -dti */ .*/	Prints all non hidden and hidden folders in the current working directory, including inode numbers, in chronological order, going from newest to oldest.
ls -dtri */ .*/	Prints all non hidden and hidden folders in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest.
ls -dlti */ .*/	Prints in detail all non hidden and hidden folders in the current working directory, including inode numbers, in chronological order, going from newest to oldest.
ls -dltri */ .*/	Prints in detail all non hidden and hidden folders in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest.

B. Listing Files

Non hidden files

Command	Output
ls -p grep -v /	Prints all non hidden files in the current working directory in alphabetical order.
ls -pr grep -v /	Prints all non hidden files in the current working directory in reverse alphabetical order.
ls -pl grep -v /	
ls -l grep -v ^d	Prints in detail all non hidden files in the current working directory in alphabetical order.
ls -l grep '^\\-'	
ls -plr grep -v /	
ls -lr grep -v ^d	Prints in detail all non hidden files in the current working directory in reverse alphabetical order.
ls -lr grep '^\\-'	

ls -pt grep -v /	Prints all non hidden files in the current working directory in chronological order, going from newest to oldest.
ls -ptr grep -v /	Prints all non hidden files in the current working directory in reverse chronological order, going from oldest to newest.
ls -plt grep -v /	Prints in detail all non hidden files in the current working directory in chronological order, going from newest to oldest.
ls -lt grep -v ^d	
ls -lt grep '^\\-'	
ls -pltr grep -v /	Prints in detail all non hidden files in the current working directory in reverse chronological order, going from oldest to newest.
ls -ltr grep -v ^d	
ls -lt grep '^\\-'	
ls -pi grep -v /	Prints all non hidden files in the current working directory, including inode numbers, in alphabetical order.
ls -pri grep -v /	Prints all non hidden files in the current working directory, including inode numbers, in reverse alphabetical order.
ls -pli grep -v /	Prints in detail all non hidden files in the current working directory, including inode numbers, in alphabetical order.
ls -li grep -v ^d	
ls -plri grep -v /	Prints in detail all non hidden files in the current working directory, including inode numbers, in reverse alphabetical order.
ls -lri grep -v ^d	
ls -pti grep -v /	Prints all non hidden files in the current working directory, including inode numbers, in chronological order, going from newest to oldest.
ls -ptri grep -v /	Prints all non hidden files in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest.
ls -plti grep -v /	Prints in detail all non hidden files in the current working directory, including inode numbers, in chronological order, going from newest to oldest.
ls -lti grep -v ^d	
ls -pltri grep -v /	Prints in detail all non hidden files in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest.
ls -ltri grep -v ^d	

Hidden Files

Command	Output
ls -d .?*	
ls -a grep '^\..'	Prints all hidden files in the current working directory in alphabetical order.
ls -dr .?*	
ls -ar grep '^\..'	Prints all hidden files in the current working directory in reverse alphabetical order.
ls -ld .?*	Prints in detail all hidden files in the current working directory in alphabetical order.
ls -ldr .?*	Prints in detail all hidden files in the current working directory in reverse alphabetical order.
ls -dt .?*	
ls -at grep '^\..'	Prints all hidden files in the current working directory in chronological order, going from newest to oldest.
ls -dtr .?*	
ls -atr grep '^\..'	Prints all hidden files in the current working directory in reverse chronological order, going from oldest to newest.
ls -dlt .?*	Prints in detail all hidden files in the current working directory in chronological order, going from newest to oldest.
ls -dltr .?*	Prints in detail all hidden files in the current working directory in reverse chronological order, going from oldest to newest.
ls -di .?*	Prints all hidden files in the current working directory, including inode numbers, in alphabetical order.
ls -dir .?*	Prints all hidden files in the current working directory, including inode numbers, in reverse alphabetical order.
ls -ldi .?*	Prints in detail all hidden files in the current working directory, including inode numbers, in alphabetical order.
ls -ldri .?*	Prints in detail all hidden files in the current working directory, including inode numbers, in reverse alphabetical order.
ls -dti .?*	Prints all hidden files in the current working directory, including inode numbers, in chronological order, going from newest to oldest.
ls -dri .?*	Prints all hidden files in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest.

ls -dlti .?*	Prints in detail all hidden files in the current working directory, including inode numbers, in chronological order, going from newest to oldest.
ls -dltri .?*	Prints in detail all hidden files in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest.

Non hidden and hidden files

Command	Output
ls -pa grep -v /	Prints all non hidden and hidden files in the current working directory in alphabetical order.
ls -pra grep -v /	Prints all non hidden and hidden files in the current working directory in reverse alphabetical order.
ls -pla grep -v /	Prints in detail all non hidden and hidden files in the current working directory in alphabetical order.
ls -la grep -v ^d	
ls -la grep '^\\-'	
ls -prla grep -v /	Prints in detail all non hidden and hidden files in the current working directory in reverse alphabetical order.
ls -rla grep -v ^d	
ls -lra grep '^\\-'	
ls -pta grep -v /	Prints all non hidden and hidden files in the current working directory in chronological order, going from newest to oldest.
ls -ptra grep -v /	Prints all non hidden and hidden files in the current working directory in reverse chronological order, going from oldest to newest.
ls -plta grep -v /	Prints in detail all non hidden and hidden files in the current working directory in chronological order, going from newest to oldest.
ls -lta grep -v ^d	
ls -lta grep '^\\-'	
ls -pltra grep -v /	Prints in detail all non hidden and hidden files in the current working directory in reverse chronological order, going from oldest to newest.
ls -ltra grep -v ^d	
ls -ltra grep '^\\-'	
ls -pai grep -v /	Prints all non hidden and hidden files in the current working directory, including inode numbers, in alphabetical order.
ls -prai grep -v /	Prints all non hidden and hidden files in the current working directory, including inode numbers, in reverse alphabetical order.

ls -plai grep -v /	Prints in detail all non hidden and hidden files in the current working directory, including inode numbers, in alphabetical order.
ls -prlai grep -v /	Prints in detail all non hidden and hidden files in the current working directory, including inode numbers, in reverse alphabetical order.
ls -ptia grep -v /	Prints all non hidden and hidden files in the current working directory, including inode numbers, in chronological order, going from newest to oldest.
ls -ptrai grep -v /	Prints all non hidden and hidden files in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest.
ls -pltai grep -v /	Prints in detail all non hidden and hidden files in the current working directory, including inode numbers, in chronological order, going from newest to oldest.
ls -pltrai grep -v /	Prints in detail all non hidden and hidden files in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest.

C. Listing Folders And Files

Non hidden folders and files

Command	Output
ls	Prints all non hidden folders and files in the current working directory in alphabetical order.
ls --group-directories-first	Prints all non hidden folders in the current working directory in alphabetical order, followed by all non hidden files in the current working directory in alphabetical order.
ls -r	Prints all non hidden folders and files in the current working directory in reverse alphabetical order.
ls -r --group-directories-first	Prints all non hidden folders in the current working directory in reverse alphabetical order, followed by all non hidden files in the current working directory in reverse alphabetical order.

ls -l	Prints in detail all non hidden folders and files in the current working directory in alphabetical order.
ls -l --group-directories-first	Prints in detail all non hidden folders in the current working directory in alphabetical order, followed by all non hidden files in the current working directory in alphabetical order.
ls -lr	Prints in detail all non hidden folders and files in the current working directory in reverse alphabetical order.
ls -lr --group-directories-first	Prints in detail all non hidden folders in the current working directory in reverse alphabetical order, followed by all non hidden files in the current working directory in reverse alphabetical order.
ls -t	Prints all non hidden folders and files in the current working directory in chronological order, going from newest to oldest.
ls -t --group-directories-first	Prints all non hidden folders in the current working directory in chronological order, going from newest to oldest, followed by all non hidden files in the current working directory in chronological order, going from newest to oldest.
ls -tr	Prints all non hidden folders and files in the current working directory in reverse chronological order, going from oldest to newest.
ls -tr --group-directories-first	Prints all non hidden folders in the current working directory in reverse chronological order, going from oldest to newest, followed by all non hidden files in the current working directory in reverse chronological order, going from oldest to newest.
ls -lt	Prints in detail all non hidden folders and files in the current working directory in chronological order, going from newest to oldest.
ls -lt --group-directories-first	Prints in detail all non hidden folders in the current working directory in chronological order, going from newest to oldest, followed by all non hidden files in the current working directory in chronological order, going from newest to oldest.

ls -ltr	Prints in detail all non hidden folders and files in the current working directory in reverse chronological order, going from oldest to newest.
ls -ltr --group-directories-first	Prints in detail all non hidden folders in the current working directory in reverse chronological order, going from oldest to newest, followed by all non hidden files in the current working directory in reverse chronological order, going from oldest to newest.
ls -i	Prints all non hidden folders and files in the current working directory, including inode numbers, in alphabetical order.
ls -i --group-directories-first	Prints all non hidden folders in the current working directory, including inode numbers, in alphabetical order, followed by all non hidden files in the current working directory, including inode numbers, in alphabetical order.
ls -ri	Prints all non hidden folders and files in the current working directory, including inode numbers, in reverse alphabetical order.
ls -ri --group-directories-first	Prints all non hidden folders in the current working directory, including inode numbers, in reverse alphabetical order, followed by all non hidden files in the current working directory, including inode numbers, in reverse alphabetical order.
ls -li	Prints in detail all non hidden folders and files in the current working directory, including inode numbers, in alphabetical order.
ls -li --group-directories-first	Prints in detail all non hidden folders in the current working directory, including inode numbers, in alphabetical order, followed by all non hidden files in the current working directory, including inode numbers, in alphabetical order.
ls -lri	Prints in detail all non hidden folders and files in the current working directory, including inode numbers, in reverse alphabetical order.
ls -lri --group-directories-first	Prints in detail all non hidden folders in the current working directory, including inode numbers, in reverse alphabetical order, followed by all non hidden files in the current working directory, including inode numbers, in reverse alphabetical order.

ls -ti	Prints all non hidden folders and files in the current working directory, including inode numbers, in chronological order, going from newest to oldest.
ls -ti --group-directories-first	Prints all non hidden folders in the current working directory, including inode numbers, in chronological order, going from newest to oldest, followed by all non hidden files in the current working directory, including inode numbers, in chronological order, going from newest to oldest.
ls -tri	Prints all non hidden folders and files in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest.
ls -tri --group-directories-first	Prints all non hidden folders in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest, followed by all non hidden files in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest.
ls -lti	Prints in detail all non hidden folders and files in the current working directory, including inode numbers, in chronological order, going from newest to oldest.
ls -lti --group-directories-first	Prints in detail all non hidden folders in the current working directory, including inode numbers, in chronological order, going from newest to oldest, followed by all non hidden files in the current working directory, including inode numbers, in chronological order, going from newest to oldest.
ls -ltri	Prints in detail all non hidden folders and files in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest.
ls -ltri --group-directories-first	Prints in detail all non hidden folders in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest, followed by all non hidden files in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest.

Hidden folders and files

Command	Output
ls -d .*	Prints all hidden folders and files in the current working directory in alphabetical order.
ls -d .[^.]*	Prints all hidden folders and files in the current working directory in alphabetical order. Returns an error if at least one hidden folder or at least one hidden file does not exist.
ls -d .* --group-directories-first	Prints all hidden folders in the current working directory in alphabetical order, followed by all hidden files in the current working directory in alphabetical order.
ls -d .[^.]* --group-directories-first	Prints all hidden folders in the current working directory in alphabetical order, followed by all hidden files in the current working directory in alphabetical order. Returns an error if at least one hidden folder or at least one hidden file does not exist.
ls -dr .*	Prints all hidden folders and files in the current working directory in reverse alphabetical order.
ls -dr .[^.]*	Prints all hidden folders and files in the current working directory in reverse alphabetical order. Returns an error if at least one hidden folder or at least one hidden file does not exist.
ls -dr .* --group-directories-first	Prints all hidden folders in the current working directory in reverse alphabetical order, followed by all hidden files in the current working directory in reverse alphabetical order.
ls -dr .[^.]* --group-directories-first	Prints all hidden folders in the current working directory in reverse alphabetical order, followed by all hidden files in the current working directory in reverse alphabetical order. Returns an error if at least one hidden folder or at least one hidden file does not exist.
ls -dl .*	Prints in detail all hidden folders and files in the current working directory in alphabetical order.
ls -dl .[^.]*	Prints in detail all hidden folders and files in the current working directory in alphabetical order. Returns an error if at least one hidden folder or at least one hidden file does not exist.
ls -dl .* --group-directories-first	Prints in detail all hidden folders in the current working directory in alphabetical order, followed by all hidden files in the current working directory in alphabetical order.

ls -dl .[^.]* --group-directories-first	Prints in detail all hidden folders in the current working directory in alphabetical order, followed by all hidden files in the current working directory in alphabetical order. Returns an error if at least one hidden folder or at least one hidden file does not exist.
ls -dlr .*	Prints in detail all hidden folders and files in the current working directory in reverse alphabetical order.
ls -dlr .[^.]*	Prints in detail all hidden folders and files in the current working directory in reverse alphabetical order. Returns an error if at least one hidden folder or at least one hidden file does not exist.
ls -dlr .* --group-directories-first	Prints in detail all hidden folders in the current working directory in reverse alphabetical order, followed by all hidden files in the current working directory in reverse alphabetical order.
ls -dlr .[^.]* --group-directories-first	Prints in detail all hidden folders in the current working directory in reverse alphabetical order, followed by all hidden files in the current working directory in reverse alphabetical order. Returns an error if at least one hidden folder or at least one hidden file does not exist.
ls -dt .*	Prints all hidden folders and files in the current working directory in chronological order, going from newest to oldest.
ls -dt .[^.]*	Prints all hidden folders and files in the current working directory in chronological order, going from newest to oldest. Returns an error if at least one hidden folder or at least one hidden file does not exist.
ls -dt .* --group-directories-first	Prints all hidden folders in the current working directory in chronological order, going from newest to oldest, followed by all hidden files in the current working directory in chronological order, going from newest to oldest.
ls -dt .[^.]* --group-directories-first	Prints all hidden folders in the current working directory in chronological order, going from newest to oldest, followed by all hidden files in the current working directory in chronological order, going from newest to oldest. Returns an error if at least one hidden folder or at least one hidden file does not exist.
ls -dtr .*	Prints all hidden folders and files in the current working directory in reverse chronological order, going from oldest to newest.

ls -dtr .[^.]*	Prints all hidden folders and files in the current working directory in reverse chronological order, going from oldest to newest. Returns an error if at least one hidden folder or at least one hidden file does not exist.
ls -dtr .* --group-directories-first	Prints all hidden folders in the current working directory in reverse chronological order, going from oldest to newest, followed by all hidden files in the current working directory in reverse chronological order, going from oldest to newest.
ls -dtr .[^.]* --group-directories-first	Prints all hidden folders in the current working directory in reverse chronological order, going from oldest to newest, followed by all hidden files in the current working directory in reverse chronological order, going from oldest to newest. Returns an error if at least one hidden folder or at least one hidden file does not exist.
ls -dtl .*	Prints in detail all hidden folders and files in the current working directory in chronological order, going from newest to oldest.
ls -dtl .[^.]*	Prints in detail all hidden folders and files in the current working directory in chronological order, going from newest to oldest. Returns an error if at least one hidden folder or at least one hidden file does not exist.
ls -dtl .* --group-directories-first	Prints in detail all hidden folders in the current working directory in chronological order, going from newest to oldest, followed by all hidden files in the current working directory in chronological order, going from newest to oldest.
ls -dtl .[^.]* --group-directories-first	Prints in detail all hidden folders in the current working directory in chronological order, going from newest to oldest, followed by all hidden files in the current working directory in chronological order, going from newest to oldest. Returns an error if at least one hidden folder or at least one hidden file does not exist.
ls -dtrl .*	Prints in detail all hidden folders and files in the current working directory in reverse chronological order, going from oldest to newest.
ls -dtrl .[^.]*	Prints in detail all hidden folders and files in the current working directory in reverse chronological order, going from oldest to newest. Returns an error if at least one hidden folder or at least one hidden file does not exist.

ls -dtrl .* --group-directories-first	Prints in detail all hidden folders in the current working directory in reverse chronological order, going from oldest to newest, followed by all hidden files in the current working directory in reverse chronological order, going from oldest to newest.
ls -dtrl .[^.]* --group-directories-first	Prints in detail all hidden folders in the current working directory in reverse chronological order, going from oldest to newest, followed by all hidden files in the current working directory in reverse chronological order, going from oldest to newest. Returns an error if at least one hidden folder or at least one hidden file does not exist.
ls -di .*	Prints all hidden folders and files in the current working directory, including inode numbers, in alphabetical order.
ls -di .[^.]*	Prints all hidden folders and files in the current working directory, including inode numbers, in alphabetical order. Returns an error if at least one hidden folder or at least one hidden file does not exist.
ls -di .* --group-directories-first	Prints all hidden folders in the current working directory, including inode numbers, in alphabetical order, followed by all hidden files in the current working directory, including inode numbers, in alphabetical order.
ls -di .[^.]* --group-directories-first	Prints all hidden folders in the current working directory, including inode numbers, in alphabetical order, followed by all hidden files in the current working directory, including inode numbers, in alphabetical order. Returns an error if at least one hidden folder or at least one hidden file does not exist.
ls -dri .*	Prints all hidden folders and files in the current working directory, including inode numbers, in reverse alphabetical order.
ls -dri .[^.]*	Prints all hidden folders and files in the current working directory, including inode numbers, in reverse alphabetical order. Returns an error if at least one hidden folder or at least one hidden file does not exist.
ls -dri .* --group-directories-first	Prints all hidden folders in the current working directory, including inode numbers, in reverse alphabetical order, followed by all hidden files in the current working directory, including inode numbers, in reverse alphabetical order.

ls -dri .[^.]* --group-directories-first	Prints all hidden folders in the current working directory, including inode numbers, in reverse alphabetical order, followed by all hidden files in the current working directory, including inode numbers, in reverse alphabetical order. Returns an error if at least one hidden folder or at least one hidden file does not exist.
ls -dli .*	Prints in detail all hidden folders and files in the current working directory, including inode numbers, in alphabetical order.
ls -dli .[^.]*	Prints in detail all hidden folders and files in the current working directory, including inode numbers, in alphabetical order. Returns an error if at least one hidden folder or at least one hidden file does not exist.
ls -dli .* --group-directories-first	Prints in detail all hidden folders in the current working directory, including inode numbers, in alphabetical order, followed by all hidden files in the current working directory, including inode numbers, in alphabetical order.
ls -dli .[^.]* --group-directories-first	Prints in detail all hidden folders in the current working directory, including inode numbers, in alphabetical order, followed by all hidden files in the current working directory, including inode numbers, in alphabetical order. Returns an error if at least one hidden folder or at least one hidden file does not exist.
ls -dlri .*	Prints in detail all hidden folders and files in the current working directory, including inode numbers, in reverse alphabetical order.
ls -dlri .[^.]*	Prints in detail all hidden folders and files in the current working directory, including inode numbers, in reverse alphabetical order. Returns an error if at least one hidden folder or at least one hidden file does not exist.
ls -dlri .* --group-directories-first	Prints in detail all hidden folders in the current working directory, including inode numbers, in reverse alphabetical order, followed by all hidden files in the current working directory, including inode numbers, in reverse alphabetical order.
ls -dlri .[^.]* --group-directories-first	Prints in detail all hidden folders in the current working directory, including inode numbers, in reverse alphabetical order, followed by all hidden files in the current working directory, including inode numbers, in reverse alphabetical order. Returns an error if at least one hidden folder or at least one hidden file does not exist.

<code>ls -dti .*</code>	Prints all hidden folders and files in the current working directory, including inode numbers, in chronological order, going from newest to oldest.
<code>ls -dti .[^.]*</code>	Prints all hidden folders and files in the current working directory, including inode numbers, in chronological order, going from newest to oldest. Returns an error if at least one hidden folder or at least one hidden file does not exist.
<code>ls -dti .* --group-directories-first</code>	Prints all hidden folders in the current working directory, including inode numbers, in chronological order, going from newest to oldest, followed by all hidden files in the current working directory, including inode numbers, in chronological order, going from newest to oldest.
<code>ls -dti .[^.]* --group-directories-first</code>	Prints all hidden folders in the current working directory, including inode numbers, in chronological order, going from newest to oldest, followed by all hidden files in the current working directory, including inode numbers, in chronological order, going from newest to oldest. Returns an error if at least one hidden folder or at least one hidden file does not exist.
<code>ls -dtri .*</code>	Prints all hidden folders and files in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest.
<code>ls -dtri .[^.]*</code>	Prints all hidden folders and files in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest. Returns an error if at least one hidden folder or at least one hidden file does not exist.
<code>ls -dtri .* --group-directories-first</code>	Prints all hidden folders in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest, followed by all hidden files in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest.
<code>ls -dtri .[^.]* --group-directories-first</code>	Prints all hidden folders in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest, followed by all hidden files in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest. Returns an error if at least one hidden folder or at least one hidden file does not exist.

ls -dtli :*	Prints in detail all hidden folders and files in the current working directory, including inode numbers, in chronological order, going from newest to oldest.
ls -dtli .[^.]*	Prints in detail all hidden folders and files in the current working directory, including inode numbers, in chronological order, going from newest to oldest. Returns an error if at least one hidden folder or at least one hidden file does not exist.
ls -dtli .* --group-directories-first	Prints in detail all hidden folders in the current working directory, including inode numbers, in chronological order, going from newest to oldest, followed by all hidden files in the current working directory, including inode numbers, in chronological order, going from newest to oldest.
ls -dtli .[^.]* --group-directories-first	Prints in detail all hidden folders in the current working directory, including inode numbers, in chronological order, going from newest to oldest, followed by all hidden files in the current working directory, including inode numbers, in chronological order, going from newest to oldest. Returns an error if at least one hidden folder or at least one hidden file does not exist.
ls -dtrli :*	Prints in detail all hidden folders and files in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest.
ls -dtrli .[^.]*	Prints in detail all hidden folders and files in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest. Returns an error if at least one hidden folder or at least one hidden file does not exist.
ls -dtrli .* --group-directories-first	Prints in detail all hidden folders in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest, followed by all hidden files in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest.

ls -dtrli .[^.]* --group-directories-first	Prints in detail all hidden folders in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest, followed by all hidden files in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest. Returns an error if at least one hidden folder or at least one hidden file does not exist.
--	---

Non hidden and hidden folders and files

Command	Output
ls -a	Prints all non hidden and hidden folders and files in the current working directory in alphabetical order.
ls -a --group-directories-first	Prints all non hidden and hidden folders in the current working directory in alphabetical order, followed by all non hidden and hidden files in the current working directory in alphabetical order.
ls -ar	Prints all non hidden and hidden folders and files in the current working directory in reverse alphabetical order.
ls -ar --group-directories-first	Prints all non hidden and hidden folders in the current working directory in reverse alphabetical order, followed by all non hidden and hidden files in the current working directory in reverse alphabetical order.
ls -la	Prints in detail all non hidden and hidden folders and files in the current working directory in alphabetical order.
ls -la --group-directories-first	Prints in detail all non hidden and hidden folders in the current working directory in alphabetical order, followed by all non hidden and hidden files in the current working directory in alphabetical order.
ls -lra	Prints in detail all non hidden and hidden folders and files in the current working directory in reverse alphabetical order.
ls -lra --group-directories-first	Prints in detail all non hidden and hidden folders in the current working directory in reverse alphabetical order, followed by all non hidden and hidden files in the current working directory in reverse alphabetical order.

ls -ta	Prints all non hidden and hidden folders and files in the current working directory in chronological order, going from newest to oldest.
ls -ta --group-directories-first	Prints all non hidden and hidden folders in the current working directory in chronological order, going from newest to oldest, followed by all non hidden and hidden files in the current working directory in chronological order, going from newest to oldest.
ls -tra	Prints all non hidden and hidden folders and files in the current working directory in reverse chronological order, going from oldest to newest.
ls -tra --group-directories-first	Prints all non hidden and hidden folders in the current working directory in reverse chronological order, going from oldest to newest, followed by all non hidden and hidden files in the current working directory in reverse chronological order, going from oldest to newest.
ls -tla	Prints in detail all non hidden and hidden folders and files in the current working directory in chronological order, going from newest to oldest.
ls -tla --group-directories-first	Prints in detail all non hidden and hidden folders in the current working directory in chronological order, going from newest to oldest, followed by all non hidden and hidden files in the current working directory in chronological order, going from newest to oldest.
ls -trla	Prints in detail all non hidden and hidden folders and files in the current working directory in reverse chronological order, going from oldest to newest.
ls -trla --group-directories-first	Prints in detail all non hidden and hidden folders in the current working directory in reverse chronological order, going from oldest to newest, followed by all non hidden and hidden files in the current working directory in reverse chronological order, going from oldest to newest.
ls -ia	Prints all non hidden and hidden folders and files in the current working directory, including inode numbers, in alphabetical order.

ls -ia --group-directories-first	Prints all non hidden and hidden folders in the current working directory, including inode numbers, in alphabetical order, followed by all non hidden and hidden files in the current working directory, including inode numbers, in alphabetical order.
ls -iar	Prints all non hidden and hidden folders and files in the current working directory, including inode numbers, in reverse alphabetical order.
ls -iar --group-directories-first	Prints all non hidden and hidden folders in the current working directory, including inode numbers, in reverse alphabetical order, followed by all non hidden and hidden files in the current working directory, including inode numbers, in reverse alphabetical order.
ls -ial	Prints in detail all non hidden and hidden folders and files in the current working directory, including inode numbers, in alphabetical order.
ls -ial --group-directories-first	Prints in detail all non hidden and hidden folders in the current working directory, including inode numbers, in alphabetical order, followed by all non hidden and hidden files in the current working directory, including inode numbers, in alphabetical order.
ls -lrai	Prints in detail all non hidden and hidden folders and files in the current working directory, including inode numbers, in reverse alphabetical order.
ls -lrai --group-directories-first	Prints in detail all non hidden and hidden folders in the current working directory, including inode numbers, in reverse alphabetical order, followed by all non hidden and hidden files in the current working directory, including inode numbers, in reverse alphabetical order.
ls -tia	Prints all non hidden and hidden folders and files in the current working directory, including inode numbers, in chronological order, going from newest to oldest.
ls -tia --group-directories-first	Prints all non hidden and hidden folders in the current working directory, including inode numbers, in chronological order, going from newest to oldest, followed by all non hidden and hidden files in the current working directory, including inode numbers, in chronological order, going from newest to oldest.
ls -tiar	Prints all non hidden and hidden folders and files in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest.

ls -tiar --group-directories-first	Prints all non hidden and hidden folders in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest, followed by all non hidden and hidden files in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest.
ls -tlia	Prints in detail all non hidden and hidden folders and files in the current working directory in chronological order, going from newest to oldest.
ls -tlia --group-directories-first	Prints in detail all non hidden and hidden folders in the current working directory, including inode numbers, in chronological order, going from newest to oldest, followed by all non hidden and hidden files in the current working directory, including inode numbers, in chronological order, going from newest to oldest.
ls -tlari	Prints in detail all non hidden and hidden folders and files in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest.
ls -tlari --group-directories-first	Prints in detail all non hidden and hidden folders in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest, followed by all non hidden and hidden files in the current working directory, including inode numbers, in reverse chronological order, going from oldest to newest.

Sources:

<https://stackoverflow.com/questions/14352290/listing-only-directories-using-ls-in-bash-an-examination>

<https://serverfault.com/questions/368370/how-do-i-exclude-directories-when-listing-files>

<https://www.cyberciti.biz/faq/bash-shell-display-only-hidden-dot-files/>

<https://askubuntu.com/questions/468901/how-to-show-only-hidden-files-in-terminal>

Imran Afzal for the command: ls -a | grep '^\..'

HANDOUTS

MODULE 4

WELCOME TO: MODULE 4

LINUX FUNDAMENTALS

COMMANDS SYNTAX

- Command options and arguments

Commands typically have the syntax:

command option(s) argument(s)

Options:

Modify the way that a command works

Usually consist of a hyphen or dash followed by a single letter

Some commands accept multiple options which can usually be grouped together after a single hyphen

Arguments:

Most commands are used together with one or more arguments

Some commands assume a default argument if none is supplied

Arguments are optional for some commands and required by others

FILE PERMISSIONS

- UNIX is a multi-user system. Every file and directory in your account can be protected from or made accessible to other users by changing its access permissions. Every user has responsibility for controlling access to their files.
- Permissions for a file or directory may be restricted to by types
- There are 3 type of permissions
 - r - read
 - w - write
 - x - executable = running a program
- Each permission (rwx) can be controlled at three levels:
 - u - user = yourself
 - g - group = can be people in the same project
 - o - other = everyone on the system
- File or Directory permission can be displayed by running ls –l command
 - -rwxrwxrwx
- Command to change permission
 - chmod

Permission Using Numeric Mode

- Permission to a file and directory can also be assigned numerically

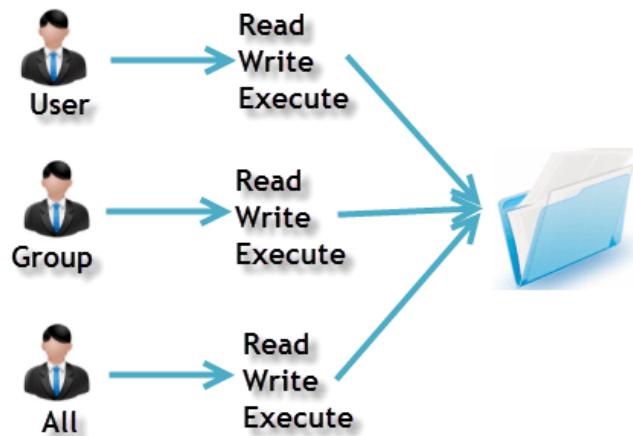
- `chmod ugo+r FILE`

OR

- `chmod 444 FILE`

-r--r--r--

Owners assigned Permission On Every File and Directory



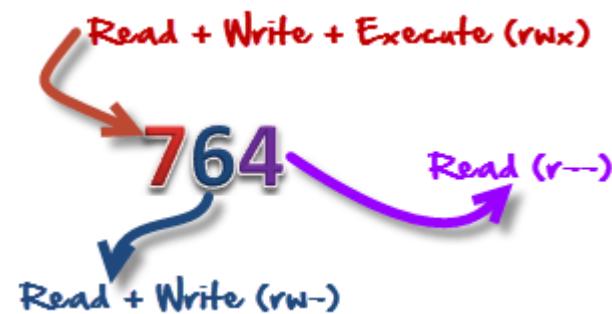
Permission Using Numeric Mode

- The table below assigns numbers to permissions types

Number	Permission Type	Symbol
0	No Permission	---
1	Execute	--x
2	Write	-w-
3	Execute + Write	-wx
4	Read	r--
5	Read + Execute	r-x
6	Read +Write	rw-
7	Read + Write +Execute	rwx

- chmod 764 FILE

764



Read + Write + Execute (rwx)

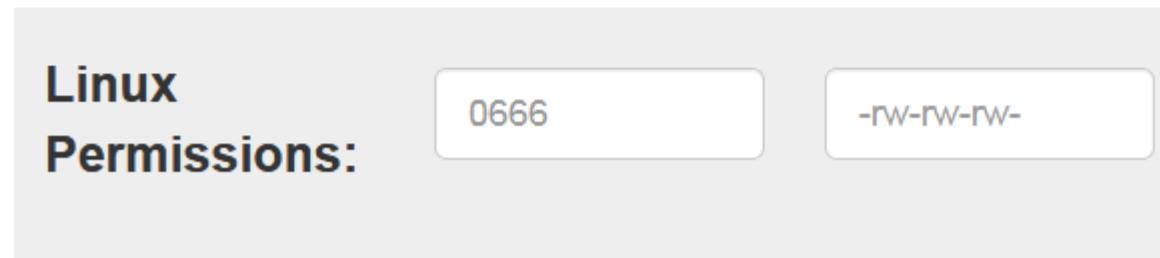
Read (r--)

Read + Write (rw-)

Permission Using Numeric Mode

- Online calculators can be used as well

Owner	Group	Public
Read <input type="checkbox"/>	Read <input type="checkbox"/>	Read <input type="checkbox"/>
Write <input type="checkbox"/>	Write <input type="checkbox"/>	Write <input type="checkbox"/>
Execute <input type="checkbox"/>	Execute <input type="checkbox"/>	Execute <input type="checkbox"/>



FILE OWNERSHIP

- There are 2 owners of a file or directory
 - User and group
- Command to change file ownership
 - chown and chgrp
 - chown changes the ownership of a file
 - chgrp changes the group ownership of a file
- Recursive ownership change option (Cascade)
 - -R

Help Commands

- There are 3 types of help commands
 - **whatis** command
 - command **--help**
 - **man** command

TAB Completion and Up Arrow

- Hitting TAB key completes the available commands, files or directories
 - **chm TAB**
 - **ls j<TAB>**
 - **cd Des<TAB>**
- Hitting up arrow key on the keyboard returns the last command ran.

Adding Text to Files (Redirects)

- 3 Simple ways to add text to a file
 - **vi**
 - **Redirect command output > or >>**
 - **echo > or >>**

INPUT AND OUTPUT REDIRECTS

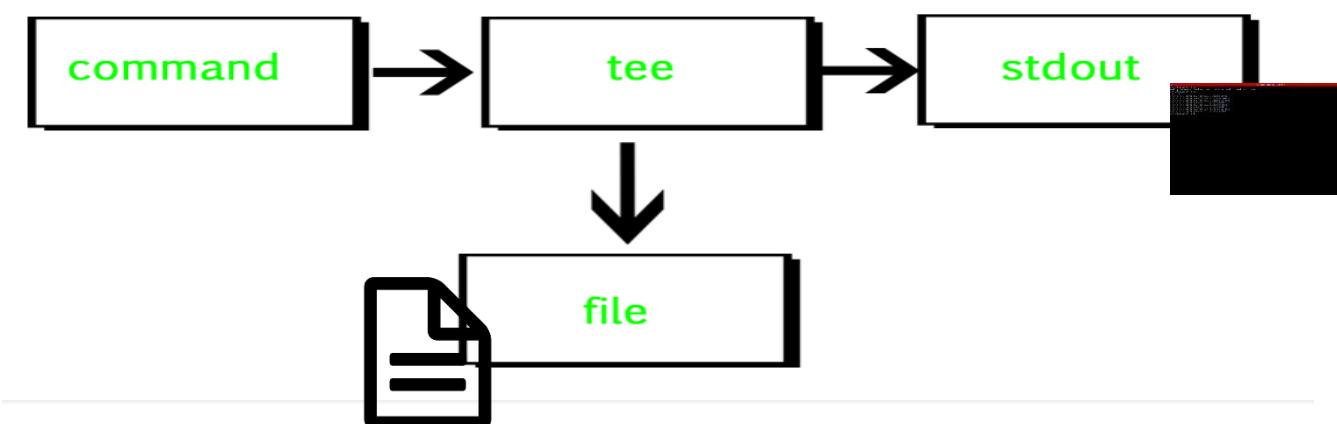
- There are 3 redirects in Linux
 1. Standard input (**stdin**) and it has file descriptor number as 0
 2. Standard output (**stdout**) and it has file descriptor number as 1
 3. Standard error (**stderr**) and it has file descriptor number as 2
- Output (**stdout**) - 1
 - By default when running a command its output goes to the terminal
 - The output of a command can be routed to a file using > symbol
 - E.g. `ls -l > listings`
`pwd > findpath`
 - If using the same file for additional output or to append to the same file then use >>
 - E.g. `ls -la >> listings`
`echo "Hello World" >> findpath.`

INPUT AND OUTPUT REDIRECTS

- Input (**stdin**) - 0
 - Input is used when feeding file contents to a file
 - E.g. **cat < listings**
mail -s "Office memo" allusers@abc.com < memoletter
- Error (**stderr**) - 2
 - When a command is executed we use a keyboard and that is also considered (stdin -0)
 - That command output goes on the monitor and that output is (stdout – 1)
 - If the command produced any error on the screen then it is considered (stderr – 2)
 - We can use redirects to route errors from the screen
 - E.g. **ls -l /root 2> errorfile**
telnet localhost 2> errorfile.

Standard Output to a File (tee)

- “tee” command is used to store and view (both at the same time) the output of any command
- The command is named after the T-splitter used in plumbing. It basically breaks the output of a program so that it can be both displayed and saved in a file. It does both the tasks simultaneously, copies the result into the specified files or variables and also display the result.

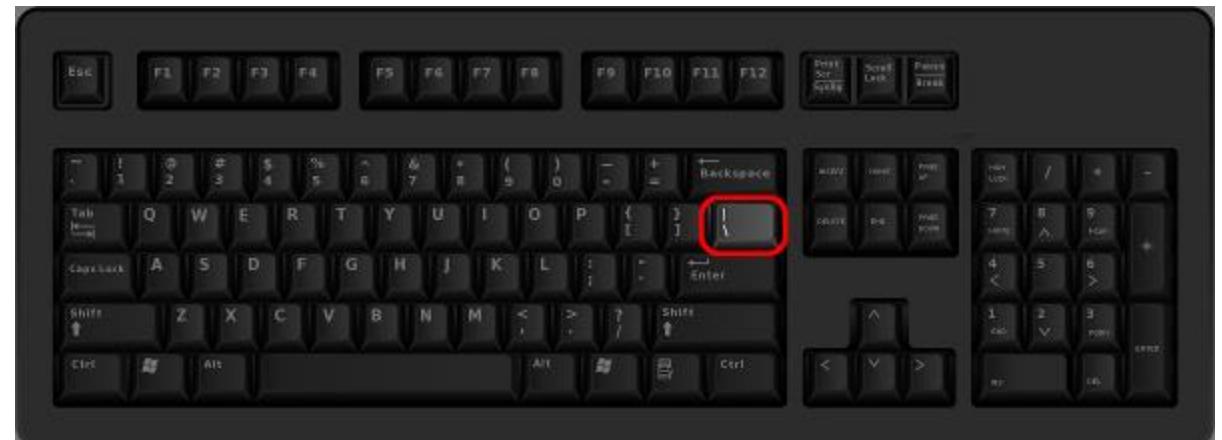


PIPES

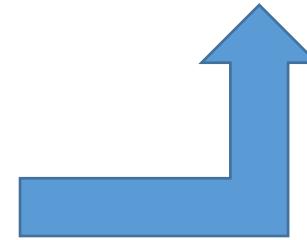
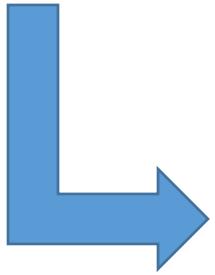
- A pipe is used by the shell to connect the output of one command directly to the input of another command.

The symbol for a pipe is the vertical bar (|). The command syntax is:

```
command1 [arguments] | command2 [arguments]
```



PIPES



`ls -l`

`|`

`more`

FILE MAINTENANCE COMMANDS

- cp
- rm
- mv
- mkdir
- rmdir or rm -r
- chgrp
- chown

FILE DISPLAY COMMANDS

- cat
- more
- less
- head
- tail

Filters / Text Processors Commands

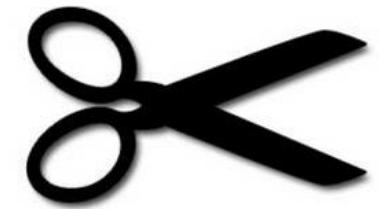
- **cut**
- **awk**
- **grep** and **egrep**
- **sort**
- **uniq**
- **wC**

cut - Text Processors Commands

cut

- cut is a command line utility that allows you to cut parts of lines from specified files or piped data and print the result to standard output. It can be used to cut parts of a line by delimiter, byte position, and character

• cut filename	=	Does not work
• cut --version	=	Check version
• cut -c1 filename	=	List one character
• cut -c1,2,4	=	Pick and chose character
• cut -c1-3 filename	=	List range of characters
• cut -c1-3,6-8 filename	=	List specific range of characters
• cut -b1-3 filename	=	List by byte size
• cut -d: -f 6 /etc/passwd	=	List first 6 th column separated by :
• cut -d: -f 6-7 /etc/passwd	=	List first 6 and 7 th column separated by :
• ls -l cut -c2-4	=	Only print user permissions of files/dir



awk - Text Processors Commands

awk

- awk is a utility/language designed for data extraction. Most of the time it is used to extract fields from a file or from an output
- **awk --version**
 - = Check version
- **awk '{print \$1}' file**
 - = List 1st field from a file
- **ls -l | awk '{print \$1,\$3}'**
 - = List 1 and 3rd field of ls -l output
- **ls -l | awk '{print \$NF}'**
 - = Last field of the output
- **awk '/Jerry/ {print}' file**
 - = Search for a specific word
- **awk -F: '{print \$1}' /etc/passwd**
 - = Ouput only 1st field of /etc/passwd
- **echo "Hello Tom" | awk '{\$2="Adam"; print \$0}'**
 - = Replace words field words
- **cat file | awk '{\$2="Imran"; print \$0}'**
 - = Replace words field words
- **awk 'length(\$0) > 15' file**
 - = Get lines that have more than 15 byte size
- **ls -l | awk '{if(\$9 == "seinfeld") print \$0;}'**
 - = Get the field matching seinfeld in /home/iafzal
- **ls -l | awk '{print NF}'**
 - = Number of fields.



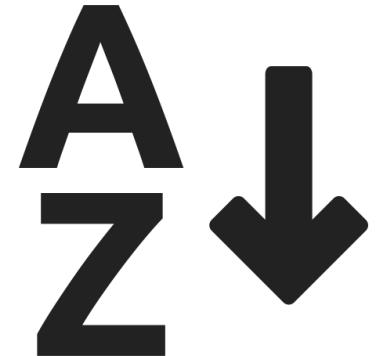
grep/egrep - Text Processors Commands

- What is grep?
 - The grep command which stands for “global regular expression print,” processes text line by line and prints any lines which match a specified pattern
- **grep --version OR grep --help** = Check version or help
- **grep keyword file** = Search for a keyword from a file
- **grep -c keyword file** = Search for a keyword and count
- **grep -i KEYword file** = Search for a keyword ignore case-sensitive
- **grep -n keyword file** = Display the matched lines and their line numbers
- **grep -v keyword file** = Display everything but keyword
- **grep keyword file | awk '{print \$1}'** = Search for a keyword and then only give the 1st field
- **ls -l | grep Desktop** = Search for a keyword and then only give the 1st field
- **egrep -i "keyword|keyword2" file** = Search for 2 keywords.



sort/uniq - Text Processors Commands

- What are sort and uniq commands?
 - Sort command sorts in alphabetical order
 - Uniq command filters out the repeated or duplicate lines
- **sort --version OR sort --help** = Check version or help
- **sort file** = Sorts file in alphabetical order
- **sort -r file** = Sort in reverse alphabetical order
- **sort -k2 file** = Sort by field number
- **uniq file** = Removes duplicates
- **sort file | uniq** = Always sort first before using uniq their line numbers
- **sort file | uniq -c** = Sort first then uniq and list count
- **sort file | uniq -d** = Only show repeated lines.



wc - Text Processors Commands

- What is **wc** command?
 - The command reads either standard input or a list of files and generates: **newline count, word count, and byte count**



- | | |
|------------------------------------|--|
| • wc --version OR wc --help | = Check version or help |
| • wc file | = Check file line count, word count and byte count |
| • wc -l file | = Get the number of lines in a file |
| • wc -w file | = Get the number of words in a file |
| • wc -b file | = Get the number of bytes in a file |
| • wc DIRECTORY | = NOT allowed |
| • ls -1 wc -l | = Number of files |
| • grep keyword wc -l | = Number of keyword lines. |

Compare Files

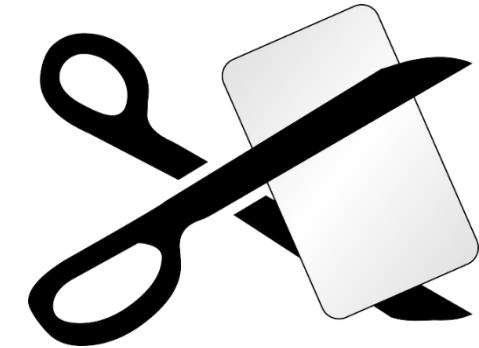
- **diff** (Line by line)
- **cmp** (Byte by byte)

Compress and un-Compress Files

- **tar**
- **gzip**
- **gzip -d OR gunzip**

Truncate File Size (`truncate`)

- The Linux `truncate` command is often used to shrink or extend the size of a file to the specified size
- Command
 - `truncate -s 10 filename`



COMBINING AND SPLITTING FILES

- Multiple files can be combined into one and
- One file can be split into multiple files

- `cat file1 file2 file3 > file4`
- `split file4`
- e.g. `split -l 300 file.txt childfile`

Split `file.txt` into 300 lines per file and output to `childfileaa`, `childfileab` and `childfileac`

Linux vs. Windows Commands

Command Description	Windows	Linux
Listing of a directory	dir	ls -l
Rename a file	ren	mv
Copy a file	copy	cp
Move file	move	mv
Clear screen	cls	clear
Delete file	del	rm
Compare contents of files	fc	diff
Search for a word/string in a file	find	grep
Display command help	command /?	man command
Displays your location in the file system	chdir	pwd
Displays the time	time	date

Linux Command Line Structure

A command is a program that tells the Linux system to do something. It has the form:

```
command [options] [arguments]
```

where an argument indicates on what the command is to perform its action, usually a file or series of files. An option modifies the command, changing the way it performs. Commands are case sensitive. command and Commands are not the same.

Options are generally preceded by a hyphen (-), and for most commands, more than one option can be strung together, in the form:

```
command -[option] [option] [option]
```

e.g.:

```
ls -alR = will perform a long list on all files in the current directory and recursively  
perform the list through all sub-directories.
```

For most commands you can separate the options, preceding each with a hyphen, e.g.:

```
command -option1 -option2 -option3  
as in: ls -a -l -R
```

Some commands have options that require parameters. Options requiring parameters are usually specified separately, e.g.:

```
lpr -Pprinter3 -# 2 file  
will send 2 copies of file to printer3.
```

These are the standard conventions for commands. However, not all Linux commands will follow the standard. Some don't require the hyphen before options and some won't let you group options together, i.e. they may require that each option be preceded by a hyphen and separated by whitespace from other options and arguments.

Options and syntax for a command are listed in the man page for the command.

File Permissions

- UNIX is a multi-user system. Every file and directory in your account can be protected from or made accessible to other users by changing its access permissions. Every user has responsibility for controlling access to their files.
 - Permissions for a file or directory may be restricted to by types
 - There are 3 type of permissions
 - r - read
 - w - write
 - x - execute = running a program
 - Each permission (rwx) can be controlled at three levels:
 - u - user = yourself
 - g - group = can be people in the same project
 - o - other = everyone on the system
 - File or Directory permission can be displayed by running ls -l command
 - -rwxrwxrwx
 - Command to change permission
 - chmod

Example:

Type	User	Group	Everyone else
	rwx	rwx	rwx

- = First dash or bit identifies the file type
 - = 2nd 3 bits defines the permission for user (file or dir owner)
 - = 3rd 3 bits defines the permission for group
 - = 4th 3 bits defines the permission for everyone else

Permissions can also be change through numerical method. Each of the permission types is represented by either a numeric equivalent:

read=4, write=2, execute=1

or a single letter:

read=r, write=w, execute=x

A permission of 4 or r would specify read permissions. If the permissions desired are read and write, the 4 (representing read) and the 2 (representing write) are added together to make a permission of 6. Therefore, a permission setting of 6 would allow read and write permissions.

Common Options

-f force (no error message is generated if the change is unsuccessful)

-R recursively descend through the directory structure and change the modes

Examples

If the permission desired for file1 is user: read, write, execute, group: read, execute, other: read, execute, the command to use would be

chmod 755 file1 or chmod u=rwx,go=rx file1

Reminder: When giving permissions to group and other to use a file, it is necessary to allow at least execute permission to the directories for the path in which the file is located. The easiest way to do this is to be in the directory for which permissions need to be granted:

chmod 711 . or chmod u=rw,+x . or chmod u=rwx,go=x .

where the dot (.) indicates this directory.

File Ownership

chown - change ownership

Ownership of a file can be changed with the chown command. On most versions of Unix this can only be done by the super-user, i.e. a normal user can't give away ownership of their files. chown is used as below, where # represents the shell prompt for the super-user:

Syntax

chown [options] user[:group] file (SVR4)

chown [options] user[.group] file (BSD)

Common Options

-R recursively descend through the directory structure
-f force, and don't report any errors

Examples

```
# chown new_owner file
```

chgrp - change group

Anyone can change the group of files they own, to another group they belong to, with the chgrp command.

Syntax

```
chgrp [options] group file
```

Common Options

-R recursively descend through the directory structure
-f force, and don't report any errors

Examples

```
% chgrp
```

Getting Help

- The "man" command
 - The "man" command man gives you access to an on-line manual which potentially contains a complete description of every command available on the system. In practice, the manual usually contains a subset of all commands.
 - man can also provide you with one line descriptions of commands which match a specified keyword
 - The online manual is divided into sections:

Section	Description
1	User Commands
2	System Commands
3	Subroutines
4	Devices
5	File Formats
6	Games
7	Miscellaneous
8	System Administration
l	Local Commands
n	New Commands

- Examples of using the man command:

To display the manual page for the cp (copy files) command:

```
man cp
```

--More--23% at the bottom left of the screen means that only 23% of the man page is displayed. Press the space bar to display more of it or type q to quit.

By default, the man page in section 1 is displayed if multiple sections exist. You can access a different section by specifying the section. For example:

```
man 8 telnetd
```

Keyword searching: use the -k option followed by the keyword. Two examples appear below.

```
man -k mail  
man -k 'copy files'
```

To view a one line description of what a command does:

```
what is more
```

will display what the "more" command does:

more, page (1) - browse or page through a text file

- who - shows who is on the system

```
who  
who am i
```

- finger - displays information about users, by name or login name

```
finger doe  
finger userid
```

Prompting completion

The following example shows how command-line completion works in Bash. Other command line shells may perform slightly differently.

First we type the first three letters of our command:

fir

Then we press **Tab ↵** and because the only command in our system that starts with "fir" is "firefox", it will be completed to:

firefox

Then we start typing the file name:

firefox i

But this time introduction-to-command-line-completion.html is not the only file in the current directory that starts with "i". The directory also contains files introduction-to-bash.html and introduction-to-firefox.html. The system can't decide which of these filenames we wanted to type, but it does know that the file must begin with "introduction-to-", so the command will be completed to:

firefox introduction-to-

Now we type "c":

firefox introduction-to-c

After pressing **Tab ↵** it will be completed to the whole filename:

firefox introduction-to-command-line-completion.html

In short we typed:

fir**Tab ↵**i**Tab ↵**c**Tab ↵**

This is just eight keystrokes, which is considerably less than 52 keystrokes we would have needed to type without using command-line completion.

Rotating completion

The following example shows how command-line completion works with rotating completion, such as Windows's CMD uses.

We follow the same procedure as for prompting completion until we have:

```
firefox i
```

We press **Tab ↩** once, with the result:

```
firefox introduction-to-bash.html
```

We press **Tab ↩** again, getting:

```
firefox introduction-to-command-line-completion.html
```

In short we typed:

```
firTab ↩iTab ↩Tab ↩
```

Adding Text to Files:

echo command

- echo "Your text goes here" > filename (To add text and create a new file)
- echo "Additional text" >> filename (To append to an existing file)

cp command

- cp exisiting-file new-filename (To copy an existing file to new file)
- cat existing-file > new-filename (cat the content of an existing file and add to new file. This command does the same as above)

vi command

- vi filename (Create a new file and enter text using vi insert mode)

Pipes

- A pipe is used by the shell to connect the stdout of one command directly to the stdin of another command.
- The symbol for a pipe is the vertical bar (|). The command syntax is:

```
command1 [arguments] | command2 [arguments]
```

- Pipes accomplish with one command what otherwise would take intermediate files and multiple commands. For example, operation 1 and operation 2 are equivalent:

```
Operation 1
    who > temp
        sort temp
```

```
Operation 2
    who | sort
```

- Pipes do not affect the contents of the input files.
- Two very common uses of a pipe are with the "more" and "grep" utilities. Some examples:

```
ls -al | more
who | more
ps ug | grep myuserid
who | grep kelly
```

File Maintenance Commands

cp

Copies files. Will overwrite unless otherwise specified. Must also have write permission in the destination directory.

Example:

```
cp sample.f sample2.f      - copies sample.f to sample2.f
cp -R dir1 dir2           - copies contents of directory dir1 to dir2
cp -i file.1 file.new     - prompts if file.new will be overwritten
cp *.txt chapt1            - copies all files with .txt suffix to directory
                             chapt1
cp /usr/doc/README ~       - copies file to your home directory
cp ~betty/index .          - copies the file "index" from user betty's
                             home directory to current directory
```

rm

Deletes/removes files or directories if file permissions permit

Example:

```
rm sample.f    - deletes sample.f
rm chap?.txt   - deletes all files with chap as the first four
                  characters of their name and with .txt as the last
                  four characters of their name
rm -i *         - deletes all files in current directory but asks
                  first for each file
rm -r /olddir   - recursively removes all files in the directory
                  olddir, including the directory itself
```

mv

Moves files. It will overwrite unless otherwise specified. Must also have write permission in the destination directory.

Example:

```
mv sample.f sample2.f      - moves sample.f to sample2.f
mv dir1 newdir/dir2        - moves contents of directory dir1 to
                             newdir/dir2
mv -i file.1 file.new      - prompts if file.new will be overwritten
mv *.txt chapt1             - moves all files with .txt suffix to
                             directory chapt1
```

mkdir

Make directory. Will create the new directory in your working directory by default.

Example:

```
mkdir /u/training/data  
mkdir data2
```

rmdir

Remove directory. Directories must be empty before you remove them.

```
rmdir project1
```

To recursively remove nested directories, use the rm command with the -r option:

```
rm -r directory_name
```

chgrp

Changes the group ownership of a file or directory.

Syntax

```
chgrp [-f] [-h] [-R] Group { File ... | Directory ... }  
chgrp -R [ -f ] [ -H | -L | -P ] Group { File... | Directory... }
```

Description

The **chgrp** command changes the group of the file or directory specified by the *File* or *Directory* parameter to the group specified by the *Group* parameter. The value of the *Group* parameter can be a group name from the group database or a numeric group ID. When a symbolic link is encountered and you have not specified the **-h** or **-P** flags, the **chgrp** command changes the group ownership of the file or directory pointed to by the link and not the group ownership of the link itself.

chown

The chown command is used to change the owner and group of files, directories and links. By default, the owner of a filesystem object is the user that created it. The group is a set of users that share the same access permissions (i.e., read, write and execute) for that object. The basic syntax for using chown to change owners is

```
chown [options] new_owner object(s)
```

new_owner is the user name or the numeric user ID (UID) of the new owner, and *object* is the name of the target file, directory or link. The ownership of any number of objects can be changed simultaneously.

For example, the following would transfer the ownership of a file named *file1* and a directory named *dir1* to a new owner named *alice*:

```
chown alice file1 dir1
```

In order to perform the above command, most systems are configured by default to require access to the root (i.e., system administrator) account, which can be obtained on a personal computer by using the su (i.e., substitute user) command. An error message will be returned in the event that the user does not have the proper permissions or that the specified new owner or target(s) does not exist (or is spelled incorrectly).

The ownership and group of a filesystem object can be confirmed by using the ls command with its -l (i.e., long) option. The owner is shown in the third column and the group in the fourth. Thus, for example, the owner and group of file1 can be seen by using the following:

```
ls -l file1
```

The basic syntax for using chown to change groups is

```
chown [options] :new_group object(s)
```

or

```
chown [options] .new_group object(s)
```

The only difference between the two versions is that the name or numeric ID of the new group is preceded directly by a colon in the former and by a dot in the latter; there is no functional difference. In this case, chown performs the same function as the chgrp (i.e., change group) command.

The owner and group can be changed simultaneously by combining the syntaxes for changing owner and group. That is, the name or UID of the new owner is followed directly (i.e., with no intervening spaces) by a period or colon, which is followed directly by the name or numeric ID of the new group, which, in turn, is followed by a space and then by the names of the target files, directories and/or links.

Thus, for example, the following would change the owner of a file named file2 to the user with the user name bob and change its group to group2:

```
chown bob:group2 file2
```

If a user name or UID is followed directly by a colon or dot but no group name is provided, then the group is changed to that user's login group. Thus, for example, the following would change the ownership of file3 to cathy and would also change that file's group to the login group of the new owner (which by default is usually the same as the new owner):

```
chown cathy: file3
```

Among chown's few options is -R, which operates on filesystem objects recursively. That is, when used on a directory, it can change the ownership and/or group of all objects within the directory tree beginning with that directory rather than just the ownership of the directory itself.

The -v (verbose) option provides information about every object processed. The -c is similar, but reports only when a change is made. The --help option displays the documentation found in the man online manual, and the --version option outputs version information

chmod

Change access permissions, change mode.

Syntax

```
chmod [Options]... Mode [,Mode]... file...
chmod [Options]... Numeric_Mode file...
chmod [Options]... --reference=RFile file...
```

Options

-f, --silent, --quiet	suppress most error messages
-v, --verbose	output a diagnostic for every file processed
-c, --changes	like verbose but report only when a change is made
--reference=Rfile	use Rfile's mode instead of MODE values
-R, --recursive	change files and directories recursively
--help	display help and exit
--version	output version information and exit

chmod changes the permissions of each given file according to mode, where mode describes the permissions to modify. Mode can be specified with octal numbers or with letters. Using letters is easier to understand for most people.

Permissions:

	Owner	Group	Other
Read			
Write			
Execute			

Numeric mode:

From one to four octal digits

Any omitted digits are assumed to be leading zeros.

The first digit = selects attributes for the set user ID (4) and set group ID (2) and save text image (1)S

The second digit = permissions for the user who owns the file: read (4), write (2), and execute (1)

The third digit = permissions for other users in the file's group: read (4), write (2), and execute (1)

The fourth digit = permissions for other users NOT in the file's group: read (4), write (2), and execute (1)

The octal (0-7) value is calculated by adding up the values for each digit

User (rwx) = 4+2+1 = 7

Group(rx) = 4+1 = 5

World (rx) = 4+1 = 5

chmod mode = 0755

Examples

chmod 400 file - Read by owner

chmod 040 file - Read by group

chmod 004 file - Read by world

chmod 200 file - Write by owner

chmod 020 file - Write by group

chmod 002 file - Write by world

chmod 100 file - execute by owner

chmod 010 file - execute by group

chmod 001 file - execute by world

To combine these, just add the numbers together:

chmod 444 file - Allow read permission to owner and group and world

chmod 777 file - Allow everyone to read, write, and execute file

Symbolic Mode

The format of a symbolic mode is a combination of the letters +-= rwxXstugo

Multiple symbolic operations can be given, separated by commas.

The full syntax is [ugo...][[+-]=[rwxXstugo...]]...][,...] but this is explained below.

A combination of the letters ugoa controls which users' access to the file will be changed:

User	letter
The user who owns it	u
Other users in the file's Group	g
Other users not in the file's group	o
All users	a

If none of these are given, the effect is as if was given, but bits that are set in the umask are not affected.

All users a is effectively user + group + others

The operator '+' causes the permissions selected to be added to the existing permissions of each file; '-' causes them to be removed; and '=' causes them to be the only permissions that the file has.

The letters 'rwxXstugo' select the new permissions for the affected users:

Permission	letter
Read	r
Write	w
Execute (or access for directories)	x
Execute only if the file is a directory (or already has execute permission for some user)	X
Set user or group ID on execution	s
Save program text on swap device	t
The permissions that the User who owns the file currently has for it	u
The permissions that other users in the file's Group have for it	g
Permissions that Other users not in the file's group have for it	o

Examples

Deny execute permission to everyone:

chmod a-x file

Allow read permission to everyone:

chmod a+r file

Make a file readable and writable by the group and others:

chmod go+rwx file

Make a shell script executable by the user/owner

\$ chmod u+x myscript.sh

Allow everyone to read, write, and execute the file and turn on the set group-ID:

chmod =rwx,g+s file

Notes:

When chmod is applied to a directory:

read = list files in the directory

write = add new files to the directory

execute = access files in the directory

chmod never changes the permissions of symbolic links. This is not a problem since the permissions of symbolic links are never used. However, for each symbolic link listed on the command line, chmod changes the permissions of the pointed-to file. In contrast, chmod ignores symbolic links encountered during recursive directory traversals

Visit chmod calculator

http://www.onlineconversion.com/html_chmod_calculator.htm

File Display Commands

cat - concatenate a file

Display the contents of a file with the concatenate command, cat.

Syntax

```
cat [options] [file]
```

Common Options

- n precede each line with a line number
- v display non-printing characters, except tabs, new-lines, and form-feeds
- e display \$ at the end of each line (prior to new-line) (when used with -v option)

Examples

```
% cat filename
```

You can list a series of files on the command line, and cat will concatenate them, starting each in turn, immediately after completing the previous one, e.g.:

```
% cat file1 file2 file3
```

more, less, and pg - page through a file

more, less, and pg let you page through the contents of a file one screenful at a time. These may not all be available on your Linux system. They allow you to back up through the previous pages and search for words, etc.

Syntax

```
more [options] [+pattern] [filename]  
less [options] [+pattern] [filename]  
pg [options] [+pattern] [filename]
```

Options

			Action
more	less	pg	
-c	-c	-c	clear display before displaying
	-i		ignore case
-w	default	default	don't exit at end of input, but prompt and wait
-lines		-lines	# of lines/screenful
+/pattern	+/pattern	+/pattern	search for the pattern

Internal Controls

more	displays (one screen at a time) the file requested
<space bar>	to view next screen
<return> or <CR>	to view one more line
q	to quit viewing the file
h	help
b	go back up one screenful
/word	search for word in the remainder of the file
	See the man page for additional options
less	similar to more; see the man page for options
pg	the SVR4 equivalent of more (page)

echo - echo a statement

The echo command is used to repeat, or echo, the argument you give it back to the standard output device. It normally ends with a line-feed, but you can specify an option to prevent this.

Syntax

echo [string]

Common Options

-n	don't print <new-line> (BSD, shell built-in)
\c	don't print <new-line> (SVR4)
\0n	where n is the 8-bit ASCII character code (SVR4)
\t	tab (SVR4)
\f	form-feed (SVR4)
\n	new-line (SVR4)
\v	vertical tab (SVR4)

Examples

% echo Hello Class or echo "Hello Class"

To prevent the line feed:

% echo -n Hello Class or echo "Hello Class \c"

where the style to use in the last example depends on the echo command in use.

The \x options must be within pairs of single or double quotes, with or without other string characters.

head - display the start of a file

head displays the head, or start, of the file.

Syntax

`head [options] file`

Common Options

<code>-n number</code>	number of lines to display, counting from the top of the file
<code>-number</code>	same as above

Examples

By default `head` displays the first 10 lines. You can display more with the "`-n number`", or "`-number`" options, e.g., to display the first 40 lines:

`% head -40 filename` or `head -n 40 filename`

more

Browses/displays files one screen at a time.

- Use `h` for help
- spacebar to page
- `b` for back
- `q` to quit
- `/string` to search for string

Example:

`more sample.f`

tail - display the end of a file

`tail` displays the tail, or end, of the file.

Syntax

`tail [options] file`

Common Options

<code>-number</code>	number of lines to display, counting from the bottom of the file
----------------------	--

Examples

The default is to display the last 10 lines, but you can specify different line or byte numbers, or a different starting point within the file. To display the last 30 lines of a file use the `-number` style:

`% tail -30 filename`

Filter / Text Processing Commands

grep, awk, sed

grep

The grep utility is used to search for generalized regular expressions occurring in Linux files. Regular expressions, such as those shown above, are best specified in apostrophes (or single quotes) when specified in the grep utility. The egrep utility provides searching capability using an extended set of meta-characters. The syntax of the grep utility, some of the available options, and a few examples are shown below.

Syntax

```
grep [options] regexp [file[s]]
```

Common Options

- i ignore case
- c report only a count of the number of lines containing matches, not the matches themselves
- v invert the search, displaying only lines that do not match
- n display the line number along with the line on which a match was found
- s work silently, reporting only the final status:
 - 0, for match(es) found
 - 1, for no matches
 - 2, for errors
- l list filenames, but not lines, in which matches were found

Examples

Consider the following file:

```
cat num.list
1 15 fifteen
2 14 fourteen
3 13 thirteen
4 12 twelve
5 11 eleven
6 10 ten
8 8 eight
9 7 seven
10 6 six
11 5 five
14 2 two
15 1 one
```

Here are some grep examples using this file. In the first we'll search for the number 15:

```
> grep '15' num.list  
1 15 fifteen  
15 1 one
```

Now we'll use the "-c" option to count the number of lines matching the search criterion:

```
> grep -c '15' num.list  
2
```

Here we'll be a little more general in our search, selecting for all lines containing the character 1 followed by either of 1, 2 or 5:

```
> grep '1[125]' num.list  
1 15 fifteen  
4 12 twelve  
5 11 eleven  
11 5 five  
12 4 four  
15 1 one
```

Now we'll search for all lines that begin with a space:

```
> grep '^' num.list  
1 15 fifteen  
2 14 fourteen  
3 13 thirteen  
4 12 twelve  
5 11 eleven  
6 10 ten  
7 9 nine  
8 8 eight  
9 7 seven
```

Or all lines that don't begin with a space:

```
> grep '^[^ ]' num.list  
10 6 six  
11 5 five  
12 4 four  
13 3 three  
14 2 two  
15 1 one
```

The latter could also be done by using the -v option with the original search string, e.g.:

```
> grep -v '^' num.list
```

```
10 6 six
11 5 five
12 4 four
13 3 three
14 2 two
15 1 one
```

Here we search for all lines that begin with the characters 1 through 9:

```
> grep '^[1-9]' num.list
10 6 six
11 5 five
12 4 four
13 3 three
14 2 two
15 1 one
```

This example will search for any instances of t followed by zero or more occurrences of e:

```
> grep 'te*' num.list
1 15 fifteen
2 14 fourteen
3 13 thirteen
4 12 twelve
6 10 ten
8 8 eight
13 3 three
14 2 two
```

This example will search for any instances of t followed by one or more occurrences of e:

```
> grep 'tee*' num.list
1 15 fifteen
2 14 fourteen
3 13 thirteen
6 10 ten
```

We can also take our input from a program, rather than a file. Here we report on any lines output by the who program that begin with the letter l.

```
> who | grep '^l'
lcondron tttyp0 Dec 1 02:41 (lcondron-pc.acs.)
```

sed

The non-interactive, stream editor, sed, edits the input stream, line by line, making the specified changes, and sends the result to standard output.

Syntax

`sed [options] edit_command [file]`

The format for the editing commands are:

`[address1[,address2]][function][arguments]`

where the addresses are optional and can be separated from the function by spaces or tabs. The function is required. The arguments may be optional or required, depending on the function in use.

Line-number Addresses are decimal line numbers, starting from the first input line and incremented by one for each. If multiple input files are given the counter continues cumulatively through the files. The last input line can be specified with the "\$" character.

Context Addresses are the regular expression patterns enclosed in slashes (/).

Commands can have 0, 1, or 2 comma-separated addresses with the following affects:

# of addresses	lines affected
0	every line of input
1	only lines matching the address
2	first line matching the first address and all lines until, and including, the line matching the second address. The process is then repeated on subsequent lines.

Substitution functions allow context searches and are specified in the form:

`s/regular_expression_pattern/replacement_string/flag`

and should be quoted with single quotes ('') if additional options or functions are specified. These patterns are identical to context addresses, except that while they are normally enclosed in slashes (/), any normal character is allowed to function as the delimiter, other than <space> and <newline>.

The replacement string is not a regular expression pattern; characters do not have special meanings here, except:

&	substitute the string specified by regular_expression_pattern
\n	substitute the nth string matched by regular_expression_pattern enclosed in '\(, '\)' pairs.

These special characters can be escaped with a backslash (\) to remove their special meaning

Common Options

- e script edit script
- n don't print the default output, but only those lines specified by p or s//p functions
- f script_file take the edit scripts from the file, script_file

Valid flags on the substitution functions include:

- d delete the pattern
- g globally substitute the pattern
- p print the line

Examples

This example changes all incidents of a comma (,) into a comma followed by a space (,) when doing output:

```
% cat filey | sed s/,/\ /g
```

The following example removes all incidents of Jr preceded by a space (Jr) in filey:

```
% cat filey | sed s/>\ Jr//g
```

To perform multiple operations on the input precede each operation with the -e (edit) option and quote the strings. For example, to filter for lines containing "Date: " and "From: " and replace these without the colon (:), try:

```
sed -e 's/Date: /Date /' -e 's/From: /From /'
```

To print only those lines of the file from the one beginning with "Date:" up to, and including, the one beginning with "Name:" try:

```
sed -n '/^Date:/,/^Name:/p'
```

To print only the first 10 lines of the input (a replacement for head):

```
sed -n 1,10p
```

awk, nawk, gawk

awk is a pattern scanning and processing language. Its name comes from the last initials of the three authors: Alfred. V. Aho, Brian. W. Kernighan, and Peter. J. Weinberger. nawk is new awk, a newer version of the program, and gawk is gnu awk, from the Free Software Foundation. Each version is a little different. Here we'll confine ourselves to simple examples which should be the same for all versions. On some OSs awk is really nawk.

awk searches its input for patterns and performs the specified operation on each line, or fields of the line, that contain those patterns. You can specify the pattern matching statements for awk either on

the command line, or by putting them in a file and using the -f program_file option.

Syntax

awk program [file]

where program is composed of one or more:

pattern { action }

fields. Each input line is checked for a pattern match with the indicated action being taken on a match. This continues through the full sequence of patterns, then the next line of input is checked.

Input is divided into records and fields. The default record separator is <newline>, and the variable NR keeps the record count. The default field separator is whitespace, spaces and tabs, and the variable NF keeps the field count. Input field, FS, and record, RS, separators can be set at any time to match any single character. Output field, OFS, and record, ORS, separators can also be changed to any single character, as desired. \$n, where n is an integer, is used to represent the nth field of the input record, while \$0 represents the entire input record.

BEGIN and END are special patterns matching the beginning of input, before the first field is read, and the end of input, after the last field is read, respectively.

Printing is allowed through the print, and formatted print, printf, statements.

Patterns may be regular expressions, arithmetic relational expressions, string-valued expressions, and boolean combinations of any of these. For the latter the patterns can be combined with the boolean operators below, using parentheses to define the combination:

|| or
&& and
! not

Comma separated patterns define the range for which the pattern is applicable, e.g.:

/first/,/last/

selects all lines starting with the one containing first, and continuing inclusively, through the one containing last.

To select lines 15 through 20 use the pattern range:

NR == 15, NR == 20

Regular expressions must be enclosed with slashes (/) and meta-characters can be escaped with the backslash (\). Regular expressions can be grouped with the operators:

| or, to separate alternatives

- + one or more
- ? zero or one

A regular expression match can be either of:

- ~ contains the expression
- !~ does not contain the expression

So the program:

```
$1 ~ /[Ff]rank/
```

is true if the first field, \$1, contains "Frank" or "frank" anywhere within the field. To match a field identical to "Frank" or "frank" use:

```
$1 ~ /^[Ff]rank$/
```

Relational expressions are allowed using the relational operators:

- < less than
- <= less than or equal to
- == equal to
- >= greater than or equal to
- != not equal to
- > greater than

Offhand you don't know if variables are strings or numbers. If neither operand is known to be numeric, than string comparisons are performed. Otherwise, a numeric comparison is done. In the absence of any information to the contrary, a string comparison is done, so that:

```
$1 > $2
```

will compare the string values. To ensure a numerical comparison do something similar to:

```
( $1 + 0 ) > $2
```

The mathematical functions: exp, log and sqrt are built-in

Some other built-in functions include:

- | | |
|---------------|---|
| index(s,t) | returns the position of string s where t first occurs, or 0 if it doesn't |
| length(s) | returns the length of string s |
| substr(s,m,n) | returns the n-character substring of s, beginning at position m |

Arrays are declared automatically when they are used, e.g.:

```
arr[i] = $1
```

assigns the first field of the current input record to the ith element of the array.

Flow control statements using if-else, while, and for are allowed with C type syntax:

```
for (i=1; i <= NF; i++) {actions}
```

```
while (i<=NF) {actions}
if (i<NF) {actions}
```

Common Options

- f program_file read the commands from program_file
- Fc use character c as the field separator character

Examples

```
% cat filex | tr a-z A-Z | awk -F: '{printf ("7R %-6s %-9s %-24s \n",$1,$2,$3)}'>upload.file
```

cats filex, which is formatted as follows:

```
nfb791:99999999:smith
7ax791:99999999:jones
8ab792:99999999:chen
8aa791:99999999:mcnulty
```

changes all lower case characters to upper case with the tr utility, and formats the file into the following which is written into the file upload.file:

```
7R NFB791 99999999 SMITH
7R 7AX791 99999999 JONES
7R 8AB792 99999999 CHEN
7R 8AA791 99999999 MCNULTY
```

cut - select parts of a line

The *cut* command allows a portion of a file to be extracted for another use.

Syntax

```
cut [options] file
```

Common Options

-c character_list character positions to select (first character is 1)

-d delimiter field delimiter (defaults to <TAB>)

-f field_list fields to select (first field is 1)

Both the character and field lists may contain comma-separated or blank-character-separated numbers (in increasing order), and may contain a hyphen (-) to indicate a range. Any numbers missing at either before (e.g. -5) or after (e.g. 5-) the hyphen indicates the full range starting with the first, or ending with the last character or field, respectively. Blank-character-separated lists must be enclosed in quotes. The field delimiter should be enclosed in quotes if it has special meaning to the shell, e.g. when specifying a <space> or <TAB> character.

Examples

In these examples we will use the file **users**:

```
jdoe John Doe 4/15/96
lsmith Laura Smith 3/12/96
```

```
pchen Paul Chen 1/5/96
jhsu Jake Hsu 4/17/96
sphilip Sue Phillip 4/2/96
```

If you only wanted the username and the user's real name, the *cut* command could be used to get only that information:

```
% cut -f 1,2 users
jdoe John Doe
lsmith Laura Smith
pchen Paul Chen
jhsu Jake Hsu
sphilip Sue Phillip
```

The *cut* command can also be used with other options. The **-c** option allows characters to be the selected cut. To select the first 4 characters:

```
% cut -c 1-4 users
This yields:
jdoe
lsmi
pcbe
jhsu
sphi
thus cutting out only the first 4 characters of each line.
```

paste - merge files

The *paste* command allows two files to be combined side-by-side. The default delimiter between the columns in a paste is a tab, but options allow other delimiters to be used.

Syntax

```
paste [options] file1 file2
```

Common Options

-d list list of delimiting characters

-s concatenate lines

The list of **delimiters** may include a single character such as a comma; a quoted string, such as a space; or any of the following escape sequences:

\n <newline> character

\t <tab> character

\\\ backslash character

\0 empty string (non-null character)

It may be necessary to quote delimiters with special meaning to the shell.

A hyphen (-) in place of a file name is used to indicate that field should come from standard input.

Examples

Given the file **users**:

```
jdoe John Doe 4/15/96
lsmith Laura Smith 3/12/96
pchen Paul Chen 1/5/96
jhsu Jake Hsu 4/17/96
sphilip Sue Phillip 4/2/96
and the file phone:
John Doe 555-6634
Laura Smith 555-3382
Paul Chen 555-0987
Jake Hsu 555-1235
Sue Phillip 555-7623
```

the **paste** command can be used in conjunction with the **cut** command to create a new file, **listing**, that includes the username, real name, last login, and phone number of all the users. First, extract the phone numbers into a temporary file, **temp.file**:

```
% cut -f2 phone > temp.file
555-6634
555-3382
555-0987
555-1235
555-7623
```

The result can then be pasted to the end of each line in **users** and directed to the new file, **listing**:

```
% paste users temp.file > listing
jdoe John Doe 4/15/96 237-6634
lsmith Laura Smith 3/12/96 878-3382
pchen Paul Chen 1/5/96 888-0987
jhsu Jake Hsu 4/17/96 545-1235
sphilip Sue Phillip 4/2/96 656-7623
```

This could also have been done on one line without the temporary file as:

```
% cut -f2 phone | paste users - > listing
```

with the same results. In this case the hyphen (-) is acting as a placeholder for an input field (namely, the output of the **cut** command).

sort - sort file contents

The **sort** command is used to order the lines of a file. Various options can be used to choose the order as well as the field on which a file is sorted. Without any options, the sort compares entire lines in the file and outputs them in ASCII order (numbers first, upper case letters, then lower case letters).

Syntax

```
sort [options] [+pos1 [-pos2]] file
```

Common Options

-b ignore leading blanks (<space> & <tab>) when determining starting and ending characters for the sort key

-d dictionary order, only letters, digits, <space> and <tab> are significant

-f fold upper case to lower case

-k keydef sort on the defined keys (not available on all systems)
-i ignore non-printable characters
-n numeric sort
-o outfile output file
-r reverse the sort
-t char use char as the field separator character
-u unique; omit multiple copies of the same line (after the sort)
+pos1 [-pos2] (old style) provides functionality similar to the "-k keydef" option.

For the **+/-position** entries **pos1** is the starting word number, beginning with **0** and **pos2** is the ending word number. When **-pos2** is omitted the sort field continues through the end of the line. Both **pos1** and **pos2** can be written in the form **w.c**, where **w** is the word number and **c** is the character within the word. For **c 0** specifies the delimiter preceding the first character, and **1** is the first character of the word. These entries can be followed by type modifiers, e.g. **n** for numeric, **b** to skip blanks, etc.

The **keydef** field of the "**-k**" option has the syntax:

```
start_field [type] [ ,end_field [type] ]
```

where:

start_field, **end_field** define the keys to restrict the sort to a portion of the line
type modifies the sort, valid modifiers are given the single characters (bdfiMnr)
from the similar sort options, e.g. a type **b** is equivalent to "**-b**", but applies
only to the specified field

Examples

In the file **users**:

```
jdoe John Doe 4/15/96
lsmith Laura Smith 3/12/96
pchen Paul Chen 1/5/96
jhsu Jake Hsu 4/17/96
sphilip Sue Phillip 4/2/96
sort users yields the following:
jdoe John Doe 4/15/96
jhsu Jake Hsu 4/17/96
lsmith Laura Smith 3/12/96
pchen Paul Chen 1/5/96
sphilip Sue Phillip 4/2/96
```

If, however, a listing sorted by last name is desired, use the option to specify which field to sort on (fields are numbered starting at 0):

```
% sort +2 users:
pchen Paul Chen 1/5/96
jdoe John Doe 4/15/96
jhsu Jake Hsu 4/17/96
sphilip Sue Phillip 4/2/96
lsmith Laura Smith 3/12/96
```

To sort in reverse order:

```
% sort -r users:
sphilip Sue Phillip 4/2/96
pchen Paul Chen 1/5/96
lsmith Laura Smith 3/12/96
```

```
jhsu Jake Hsu 4/17/96
jdoe John Doe 4/15/96
```

A particularly useful **sort** option is the **-u** option, which eliminates any duplicate entries in a file while ordering the file. For example, the file `today.logins`:

```
sphillip
jchen
jdoe
lkeres
jmarsch
ageorge
lkeres
proy
jchen
```

shows a listing of each username that logged into the system today. If we want to know how many unique users logged into the system today, using **sort** with the **-u** option will list each user only once. (The command can then be piped into "**wc -l**" to get a number):

```
% sort -u today.logins
ageorge
jchen
jdoe
jmarsch
lkeres
proy
sphillip
```

uniq - remove duplicate lines

uniq filters duplicate adjacent lines from a file.

Syntax

```
uniq [options] [+|-n] file [file.new]
```

Common Options

- d** one copy of only the repeated lines
- u** select only the lines not repeated
- +n** ignore the first **n** characters
- s n** same as above (SVR4 only)
- n** skip the first **n** fields, including any blanks (<space> & <tab>)
- f** fields same as above (SVR4 only)

Examples

Consider the following file and example, in which **uniq** removes the 4th line from **file** and places the result in a file called **file.new**.

```
$ cat file
```

```

1 2 3 6
4 5 3 6
7 8 9 0
7 8 9 0

$ uniq file file.new

$ cat file.new
1 2 3 6
4 5 3 6
7 8 9 0

```

Below, the **-n** option of the *uniq* command is used to skip the first 2 fields in **file**, and filter out lines which are duplicates from the 3rd field onward.

```

$ uniq -2 file
1 2 3 6
7 8 9 0

```

tee - copy command output

tee sends standard in to specified files and also to standard out. It's often used in command pipelines.

Syntax

```
tee [options] [file[s]]
```

Common Options

- a append the output to the files
- i ignore interrupts

Examples

In this first example the output of who is displayed on the screen and stored in the file users.file:

```

> who | tee users.file
      condron      tttyp0      Apr 22 14:10      (lcondron-pc.acs.)
      frank        tttyp1      Apr 22 16:19      (nyssa)
      condron      tttyp9      Apr 22 15:52      (lcondron-mac.acs)

> cat users.file
      Condron      tttyp0      Apr 22 14:10      (lcondron-pc.acs.)
      Frank        tttyp1      Apr 22 16:19      (nyssa)
      Condron      tttyp9      Apr 22 15:52      (lcondron-mac.acs)

```

In this next example the output of who is sent to the files users.a and users.b. It is also piped to the wc command, which reports the line count.

```

> who | tee users.a users.b | wc -l
      3

> cat users.a
      condron      tttyp0      Apr 22 14:10      (lcondron-pc.acs.)
      frank        tttyp1      Apr 22 16:19      (nyssa)

```

```

condron    tttyp9      Apr 22 15:52      (lcondron-mac.acs)

> cat users.b
condron    tttyp0      Apr 22 14:10      (lcondron-pc.acs.)
frank     tttyp1      Apr 22 16:19      (nyssa)
condron    tttyp9      Apr 22 15:52      (lcondron-mac.acs)

```

In the following example a long directory listing is sent to the file files.long. It is also piped to the grep command which reports which files were last modified in August.

```

> ls -l | tee files.long |grep Aug
1 drwxr-sr-x 2 condron 512 Aug 8 1995 News/
2 -rw-r--r-- 1 condron 1076 Aug 8 1995 magnus.cshrc
2 -rw-r--r-- 1 condron 1252 Aug 8 1995 magnus.login

> cat files.long
total 34
2 -rw-r--r-- 1 condron 1253 Oct 10 1995 #.login#
1 drwx----- 2 condron 512 Oct 17 1995 Mail/
1 drwxr-sr-x 2 condron 512 Aug 8 1995 News/
5 -rw-r--r-- 1 condron 4299 Apr 21 00:18 editors.txt
2 -rw-r--r-- 1 condron 1076 Aug 8 1995 magnus.cshrc
2 -rw-r--r-- 1 condron 1252 Aug 8 1995 magnus.login
7 -rw-r--r-- 1 condron 6436 Apr 21 23:50 resources.txt
4 -rw-r--r-- 1 condron 3094 Apr 18 18:24 telnet.ftp
1 drwxr-sr-x 2 condron 512 Apr 21 23:56 uc/
1 -rw-r--r-- 1 condron 1002 Apr 22 00:14 uniq.tee.txt
1 -rw-r--r-- 1 condron 1001 Apr 20 15:05 uniq.tee.txt~
7 -rw-r--r-- 1 condron 6194 Apr 15 20:18 Linuxgrep.txt

```

Finding System Information:

```
uname -a  
cat /etc/redhat-release  
dmidecode
```

uname:

Sometimes it is required to quickly determine details like kernel name, version, hostname, etc of the Linux box you are using.

Even though you can find all these details in respective files present under the proc filesystem, it is easier to use uname utility to get these information quickly.

The basic syntax of the uname command is:

```
uname [OPTION]...
```

Now lets look at some examples that demonstrate the usage of ‘uname’ command.

uname without any option

When the ‘uname’ command is run without any option then it prints just the kernel name. So the output below shows that its the ‘Linux’ kernel that is used by this system.

```
$ uname  
Linux
```

You can also use uname -s, which also displays the kernel name.

```
$ uname -s  
Linux
```

Get the network node host name using -n option

Use uname -n option to fetch the network node host name of your Linux box.

```
$ uname -n  
dev-server
```

The output above will be the same as the output of the hostname command.

Get kernel release using -r option

uname command can also be used to fetch the kernel release information. The option -r can be used for this purpose.

```
$ uname -r  
2.6.32-100.28.5.el6.x86_64
```

Get the kernel version using -v option

uname command can also be used to fetch the kernel version information. The option -v can be used for this purpose.

```
$ uname -v  
#1 SMP Wed Feb 2 18:40:23 EST 2011
```

Get the machine hardware name using -m option

uname command can also be used to fetch the machine hardware name. The option -m can be used for this purpose. This indicates that it is a 64-bit system.

```
$ uname -m  
x86_64
```

Get the processor type using -p option

uname command can also be used to fetch the processor type information. The option -p can be used for this purpose. If the uname command is not able to fetch the processor type information then it produces 'unknown' in the output.

```
$ uname -p  
x86_64
```

Sometimes you might see 'unknown' as the output of this command, if uname was not able to fetch the information on processor type.

Get the hardware platform using -i option

uname command can also be used to fetch the hardware platform information. The option -i can be used for this purpose. If the uname command is not able to fetch the hardware platform information then it produces 'unknown' in the output.

```
$ uname -i  
x86_64
```

Sometimes you might see 'unknown' as the output of this command, if uname was not able to fetch the information about the platform.

Get the operating system name using the -o option

uname command can also be used to fetch the operating system name. The option -o can be used for this purpose.

For example :

```
$ uname -o  
GNU/Linux
```

cat /etc/redhat-release:

- This file provides information about your system distribution and its version
- You can also run /etc/*rel* for systems that are not on CentOS or Redhat

Dmidecode:

dmidecode is a tool for dumping a computer's DMI (some say SMBIOS) table contents in a human-readable format. This table contains a description of the system's hardware components, as well as other useful pieces of information such as serial numbers and BIOS revision. Thanks to this table, you can retrieve this information without having to probe for the actual hardware.

Take a look at

```
man dmidecode
```

to find out all options. The most common option is the --type switch which takes one or more of the following keywords:

bios, system, baseboard, chassis, processor, memory, cache, connector, slot

You can as well specify one or more of the following numbers:

Type	Information
<hr/>	
0	BIOS
1	System

- 2 Base Board
- 3 Chassis
- 4 Processor
- 5 Memory Controller
- 6 Memory Module
- 7 Cache
- 8 Port Connector
- 9 System Slots
- 10 On Board Devices
- 11 OEM Strings
- 12 System Configuration Options
- 13 BIOS Language
- 14 Group Associations
- 15 System Event Log
- 16 Physical Memory Array
- 17 Memory Device
- 18 32-bit Memory Error
- 19 Memory Array Mapped Address
- 20 Memory Device Mapped Address
- 21 Built-in Pointing Device
- 22 Portable Battery
- 23 System Reset
- 24 Hardware Security
- 25 System Power Controls
- 26 Voltage Probe
- 27 Cooling Device
- 28 Temperature Probe
- 29 Electrical Current Probe
- 30 Out-of-band Remote Access
- 31 Boot Integrity Services
- 32 System Boot
- 33 64-bit Memory Error
- 34 Management Device
- 35 Management Device Component
- 36 Management Device Threshold Data
- 37 Memory Channel
- 38 IPMI Device
- 39 Power Supply

Each keyword is equivalent to a list of type numbers:

Keyword	Types
---------	-------

bios	0, 13
------	-------

system 1, 12, 15, 23, 32
baseboard 2, 10
chassis 3
processor 4
memory 5, 6, 16, 17
cache 7
connector 8
slot 9

Here are a few sample outputs from one of my servers:

```
dmidecode --type bios
```

```
server1:/home/admin# dmidecode --type bios
# dmidecode 2.8
SMBIOS 2.5 present.
```

Handle 0x0000, DMI type 0, 24 bytes

BIOS Information

Vendor: American Megatrends Inc.

Version: V1.5B2

Release Date: 10/31/2007

Address: 0xF0000

Runtime Size: 64 kB

ROM Size: 1024 kB

Characteristics:

ISA is supported

PCI is supported

PNP is supported

APM is supported

BIOS is upgradeable

BIOS shadowing is allowed

ESCD support is available

Boot from CD is supported

Selectable boot is supported

BIOS ROM is socketed

EDD is supported

5.25"/1.2 MB floppy services are supported (int 13h)

3.5"/720 KB floppy services are supported (int 13h)

3.5"/2.88 MB floppy services are supported (int 13h)

Print screen service is supported (int 5h)

8042 keyboard services are supported (int 9h)
Serial services are supported (int 14h)
Printer services are supported (int 17h)
CGA/mono video services are supported (int 10h)
ACPI is supported
USB legacy is supported
LS-120 boot is supported
ATAPI Zip drive boot is supported
BIOS boot specification is supported
Targeted content distribution is supported
BIOS Revision: 8.14

Handle 0x0028, DMI type 13, 22 bytes

BIOS Language Information

Installable Languages: 1
en|US|iso8859-1

Currently Installed Language: en|US|iso8859-1

server1:/home/admin#

dmidecode --type system

server1:/home/admin# dmidecode --type system
dmidecode 2.8
SMBIOS 2.5 present.

Handle 0x0001, DMI type 1, 27 bytes

System Information

Manufacturer: MICRO-STAR INTERANTIONAL CO.,LTD
Product Name: MS-7368
Version: 1.0
Serial Number: To Be Filled By O.E.M.
UUID: Not Present
Wake-up Type: Power Switch
SKU Number: To Be Filled By O.E.M.
Family: To Be Filled By O.E.M.

Handle 0x0027, DMI type 12, 5 bytes

System Configuration Options

Option 1: To Be Filled By O.E.M.

```
server1:/home/admin#
```

```
dmidecode --type baseboard
```

```
server1:/home/admin# dmidecode --type baseboard
# dmidecode 2.8
SMBIOS 2.5 present.
```

Handle 0x0002, DMI type 2, 15 bytes

Base Board Information

Manufacturer: MICRO-STAR INTERANTIONAL CO.,LTD

Product Name: MS-7368

Version: 1.0

Serial Number: To be filled by O.E.M.

Asset Tag: To Be Filled By O.E.M.

Features:

Board is a hosting board

Board is replaceable

Location In Chassis: To Be Filled By O.E.M.

Chassis Handle: 0x0003

Type: Motherboard

Contained Object Handles: 0

Handle 0x0025, DMI type 10, 6 bytes

On Board Device Information

Type: Video

Status: Enabled

Description: To Be Filled By O.E.M.

```
server1:/home/admin#
```

```
dmidecode --type chassis
```

```
server1:/home/admin# dmidecode --type chassis
# dmidecode 2.8
SMBIOS 2.5 present.
```

Handle 0x0003, DMI type 3, 21 bytes

Chassis Information

Manufacturer: To Be Filled By O.E.M.

Type: Desktop

Lock: Not Present

Version: To Be Filled By O.E.M.
Serial Number: To Be Filled By O.E.M.
Asset Tag: To Be Filled By O.E.M.
Boot-up State: Safe
Power Supply State: Safe
Thermal State: Safe
Security Status: None
OEM Information: 0x00000000
Height: Unspecified
Number Of Power Cords: 1
Contained Elements: 0

server1:/home/admin#

dmidecode --type processor

server1:/home/admin# dmidecode --type processor
dmidecode 2.8
SMBIOS 2.5 present.

Handle 0x0004, DMI type 4, 40 bytes
Processor Information
 Socket Designation: CPU 1
 Type: Central Processor
 Family: Other
 Manufacturer: AMD
 ID: B2 0F 06 00 FF FB 8B 17
 Version: AMD Athlon(tm) 64 X2 Dual Core Processor 5600+
 Voltage: 1.5 V
 External Clock: 200 MHz
 Max Speed: 2800 MHz
 Current Speed: 2900 MHz
 Status: Populated, Enabled
 Upgrade: Other
 L1 Cache Handle: 0x0005
 L2 Cache Handle: 0x0006
 L3 Cache Handle: 0x0007
 Serial Number: To Be Filled By O.E.M.
 Asset Tag: To Be Filled By O.E.M.
 Part Number: To Be Filled By O.E.M.

server1:/home/admin#

dmidecode --type memory

```
server1:/home/admin# dmidecode --type memory
# dmidecode 2.8
SMBIOS 2.5 present.
```

Handle 0x0008, DMI type 5, 20 bytes

Memory Controller Information

- Error Detecting Method: 64-bit ECC
- Error Correcting Capabilities:
 - None
- Supported Interleave: One-way Interleave
- Current Interleave: One-way Interleave
- Maximum Memory Module Size: 512 MB
- Maximum Total Memory Size: 1024 MB
- Supported Speeds:
 - 70 ns
 - 60 ns
- Supported Memory Types:
 - SIMM
 - DIMM
 - SDRAM
- Memory Module Voltage: 3.3 V
- Associated Memory Slots: 2
 - 0x0009
 - 0x000A
- Enabled Error Correcting Capabilities:
 - None

Handle 0x0009, DMI type 6, 12 bytes

Memory Module Information

- Socket Designation: DIMM0
- Bank Connections: 0 5
- Current Speed: 161 ns
- Type: ECC DIMM
- Installed Size: 1024 MB (Double-bank Connection)
- Enabled Size: 1024 MB (Double-bank Connection)
- Error Status: OK

Handle 0x000A, DMI type 6, 12 bytes

Memory Module Information

- Socket Designation: DIMM1
- Bank Connections: 0 5
- Current Speed: 163 ns
- Type: ECC DIMM

Installed Size: 1024 MB (Double-bank Connection)
Enabled Size: 1024 MB (Double-bank Connection)
Error Status: OK

Handle 0x0029, DMI type 16, 15 bytes

Physical Memory Array

Location: System Board Or Motherboard
Use: System Memory
Error Correction Type: None
Maximum Capacity: 8 GB
Error Information Handle: Not Provided
Number Of Devices: 2

Handle 0x002B, DMI type 17, 27 bytes

Memory Device

Array Handle: 0x0029
Error Information Handle: Not Provided
Total Width: 64 bits
Data Width: 72 bits
Size: 1024 MB
Form Factor: DIMM
Set: None
Locator: DIMM0
Bank Locator: BANK0
Type: DDR2
Type Detail: Synchronous
Speed: 333 MHz (3.0 ns)
Manufacturer: Manufacturer0
Serial Number: SerNum0
Asset Tag: AssetTagNum0
Part Number: PartNum0

Handle 0x002D, DMI type 17, 27 bytes

Memory Device

Array Handle: 0x0029
Error Information Handle: Not Provided
Total Width: 64 bits
Data Width: 72 bits
Size: 1024 MB
Form Factor: DIMM
Set: None
Locator: DIMM1
Bank Locator: BANK1
Type: DDR2

Type Detail: Synchronous
Speed: 333 MHz (3.0 ns)
Manufacturer: Manufacturer1
Serial Number: SerNum1
Asset Tag: AssetTagNum1
Part Number: PartNum1

server1:/home/admin#

dmidecode --type cache

server1:/home/admin# dmidecode --type cache
dmidecode 2.8
SMBIOS 2.5 present.

Handle 0x0005, DMI type 7, 19 bytes

Cache Information

Socket Designation: L1-Cache
Configuration: Enabled, Not Socketed, Level 1
Operational Mode: Varies With Memory Address
Location: Internal
Installed Size: 256 KB
Maximum Size: 256 KB
Supported SRAM Types:
 Pipeline Burst
Installed SRAM Type: Pipeline Burst
Speed: Unknown
Error Correction Type: Single-bit ECC
System Type: Data
Associativity: 4-way Set-associative

Handle 0x0006, DMI type 7, 19 bytes

Cache Information

Socket Designation: L2-Cache
Configuration: Enabled, Not Socketed, Level 2
Operational Mode: Varies With Memory Address
Location: Internal
Installed Size: 1024 KB
Maximum Size: 1024 KB
Supported SRAM Types:
 Pipeline Burst
Installed SRAM Type: Pipeline Burst
Speed: Unknown
Error Correction Type: Single-bit ECC

System Type: Unified
Associativity: 4-way Set-associative

Handle 0x0007, DMI type 7, 19 bytes

Cache Information

Socket Designation: L3-Cache
Configuration: Disabled, Not Socketed, Level 3
Operational Mode: Unknown
Location: Internal
Installed Size: 0 KB
Maximum Size: 0 KB
Supported SRAM Types:
 Unknown
Installed SRAM Type: Unknown
Speed: Unknown
Error Correction Type: Unknown
System Type: Unknown
Associativity: Unknown

server1:/home/admin#

dmidecode --type connector

server1:/home/admin# dmidecode --type connector
dmidecode 2.8
SMBIOS 2.5 present.

Handle 0x000B, DMI type 8, 9 bytes

Port Connector Information

Internal Reference Designator: J1A1
Internal Connector Type: None
External Reference Designator: PS2Mouse
External Connector Type: PS/2
Port Type: Mouse Port

Handle 0x000C, DMI type 8, 9 bytes

Port Connector Information

Internal Reference Designator: J1A1
Internal Connector Type: None
External Reference Designator: Keyboard
External Connector Type: PS/2
Port Type: Keyboard Port

Handle 0x000D, DMI type 8, 9 bytes

Port Connector Information

Internal Reference Designator: J2A2
Internal Connector Type: None
External Reference Designator: USB1
External Connector Type: Access Bus (USB)
Port Type: USB

Handle 0x000E, DMI type 8, 9 bytes

Port Connector Information

Internal Reference Designator: J2A2
Internal Connector Type: None
External Reference Designator: USB2
External Connector Type: Access Bus (USB)
Port Type: USB

Handle 0x000F, DMI type 8, 9 bytes

Port Connector Information

Internal Reference Designator: J4A1
Internal Connector Type: None
External Reference Designator: LPT 1
External Connector Type: DB-25 male
Port Type: Parallel Port ECP/EPP

Handle 0x0010, DMI type 8, 9 bytes

Port Connector Information

Internal Reference Designator: J2A1
Internal Connector Type: None
External Reference Designator: COM A
External Connector Type: DB-9 male
Port Type: Serial Port 16550A Compatible

Handle 0x0011, DMI type 8, 9 bytes

Port Connector Information

Internal Reference Designator: J6A1
Internal Connector Type: None
External Reference Designator: Audio Mic In
External Connector Type: Mini Jack (headphones)
Port Type: Audio Port

Handle 0x0012, DMI type 8, 9 bytes

Port Connector Information

Internal Reference Designator: J6A1
Internal Connector Type: None
External Reference Designator: Audio Line In

External Connector Type: Mini Jack (headphones)
Port Type: Audio Port

Handle 0x0013, DMI type 8, 9 bytes

Port Connector Information

Internal Reference Designator: J6B1 - AUX IN
Internal Connector Type: On Board Sound Input From CD-ROM
External Reference Designator: Not Specified
External Connector Type: None
Port Type: Audio Port

Handle 0x0014, DMI type 8, 9 bytes

Port Connector Information

Internal Reference Designator: J6B2 - CDIN
Internal Connector Type: On Board Sound Input From CD-ROM
External Reference Designator: Not Specified
External Connector Type: None
Port Type: Audio Port

Handle 0x0015, DMI type 8, 9 bytes

Port Connector Information

Internal Reference Designator: J6J2 - PRI IDE
Internal Connector Type: On Board IDE
External Reference Designator: Not Specified
External Connector Type: None
Port Type: Other

Handle 0x0016, DMI type 8, 9 bytes

Port Connector Information

Internal Reference Designator: J6J1 - SEC IDE
Internal Connector Type: On Board IDE
External Reference Designator: Not Specified
External Connector Type: None
Port Type: Other

Handle 0x0017, DMI type 8, 9 bytes

Port Connector Information

Internal Reference Designator: J4J1 - FLOPPY
Internal Connector Type: On Board Floppy
External Reference Designator: Not Specified
External Connector Type: None
Port Type: Other

Handle 0x0018, DMI type 8, 9 bytes

Port Connector Information

Internal Reference Designator: J9H1 - FRONT PNL
Internal Connector Type: 9 Pin Dual Inline (pin 10 cut)
External Reference Designator: Not Specified
External Connector Type: None
Port Type: Other

Handle 0x0019, DMI type 8, 9 bytes

Port Connector Information

Internal Reference Designator: J1B1 - CHASSIS REAR FAN
Internal Connector Type: Other
External Reference Designator: Not Specified
External Connector Type: None
Port Type: Other

Handle 0x001A, DMI type 8, 9 bytes

Port Connector Information

Internal Reference Designator: J2F1 - CPU FAN
Internal Connector Type: Other
External Reference Designator: Not Specified
External Connector Type: None
Port Type: Other

Handle 0x001B, DMI type 8, 9 bytes

Port Connector Information

Internal Reference Designator: J8B4 - FRONT FAN
Internal Connector Type: Other
External Reference Designator: Not Specified
External Connector Type: None
Port Type: Other

Handle 0x001C, DMI type 8, 9 bytes

Port Connector Information

Internal Reference Designator: J9G2 - FNT USB
Internal Connector Type: Other
External Reference Designator: Not Specified
External Connector Type: None
Port Type: Other

Handle 0x001D, DMI type 8, 9 bytes

Port Connector Information

Internal Reference Designator: J6C3 - FP AUD
Internal Connector Type: Other
External Reference Designator: Not Specified

External Connector Type: None
Port Type: Other

Handle 0x001E, DMI type 8, 9 bytes

Port Connector Information

Internal Reference Designator: J9G1 - CONFIG
Internal Connector Type: Other
External Reference Designator: Not Specified
External Connector Type: None
Port Type: Other

Handle 0x001F, DMI type 8, 9 bytes

Port Connector Information

Internal Reference Designator: J8C1 - SCSI LED
Internal Connector Type: Other
External Reference Designator: Not Specified
External Connector Type: None
Port Type: Other

Handle 0x0020, DMI type 8, 9 bytes

Port Connector Information

Internal Reference Designator: J9J2 - INTRUDER
Internal Connector Type: Other
External Reference Designator: Not Specified
External Connector Type: None
Port Type: Other

Handle 0x0021, DMI type 8, 9 bytes

Port Connector Information

Internal Reference Designator: J9G4 - ITP
Internal Connector Type: Other
External Reference Designator: Not Specified
External Connector Type: None
Port Type: Other

Handle 0x0022, DMI type 8, 9 bytes

Port Connector Information

Internal Reference Designator: J2H1 - MAIN POWER
Internal Connector Type: Other
External Reference Designator: Not Specified
External Connector Type: None
Port Type: Other

server1:/home/admin#

```
dmidecode --type slot
```

```
server1:/home/admin# dmidecode --type slot
# dmidecode 2.8
SMBIOS 2.5 present.
```

Handle 0x0023, DMI type 9, 13 bytes

System Slot Information

Designation: AGP
Type: 32-bit AGP 4x
Current Usage: In Use
Length: Short
ID: 0
Characteristics:
 3.3 V is provided
 Opening is shared
 PME signal is supported

Handle 0x0024, DMI type 9, 13 bytes

System Slot Information

Designation: PCI1
Type: 32-bit PCI
Current Usage: Available
Length: Short
ID: 1
Characteristics:
 3.3 V is provided
 Opening is shared
 PME signal is supported

Linux Permissions Cheat Sheet

I created this repository in hopes that it may be used as a helpful reference.

Permissions

Permissions on Unix and other systems like it are split into three classes.

- User
- Group
- Other

Files and directories are owned by a **user**.

Files and directories are also assigned to a **group**.

If a user is not the owner, nor a member of the group, then they are classified as **other**.

Changing permissions

In order to change permissions, we need to first understand the two notations of permissions.

1. Symbolic notation
2. Octal notation

Symbolic notation

Symbolic notation is what you'd see on the left-hand side if you ran a command like `ls -l` in a terminal.

The first character in symbolic notation indicates the *file type* and isn't related to permissions in any way. The remaining characters are in sets of three, each representing a class of permissions.

The first class is the **user** class. The second class is the **group** class. The third class is the **other** class.

Each of the three characters for a class represents the read, write and execute permissions.

- r will be displayed if reading is permitted
- w will be displayed if writing is permitted
- x will be displayed if execution is permitted
- - will be displayed in the place of r, w, and x, if the respective permission is *not* permitted

Here are some examples of symbolic notation:

- -rwxr--r--: A regular file whose **user** class has read/write/execute, **group** class has only read permissions, **other** class has only read permissions
- drw-rw-r--: A directory whose **user** class has read/write permissions, **group** class has read/write permissions, **other** class has only read permissions
- crwxrw-r--: A character special file whose **user** has read/write/execute permissions, **group** class has read/write permissions, **other** class has only read permissions

Octal notation

Octal (base-8) notation consists of at least 3 digits (sometimes 4, the left-most digit, which represents the setuid bit, the setgid bit, and the sticky bit).

Each of the three right-most digits are the sum of its component bits in the binary numeral system.

For example:

- The read bit (r in symbolic notation) adds 4 to its total
- The write bit (w in symbolic notation) adds 2 to its total
- The execute bit (x in symbolic notation) adds 1 to its total

So what number would you use if you wanted to set a permission to read and write? $4 + 2 = 6$.

Symbolic notation	Octal notation	Plain English
-rwxr--r--	0744	user class can read/write/execute; group class can read; other class can read
-rw-rw-r--	0664	user class can read/write; group class can read/write; other class can read
-rwxrwxr--	0774	user class can read/write/execute; group class can read/write/execute; other class can read
-----	0000	None of the classes have permissions
-rwx-----	0700	user class can read/write/execute; group class has no permissions; other class has no permissions
-rwxrwxrwx	0777	All classes can read/write/execute
-rw-rw-rw	0666	All classes can read/write
-r-xr-xr-x	0555	All classes can read/execute
-r--r--r--	0444	All classes can read

--wx-wx-wx	0333	All classes can write/execute
--w--w--w-	0222	All classes can write
---x--x--x	0111	All classes can execute

All together now

Let's use the examples from the symbolic notation section and show how it'd convert to octal notation

CHMOD commands

Now that we have a better understanding of permissions and what all of these letters and numbers mean, let's take

Permission (symbolic notation)	CHMOD command	Description
-rwxrwxrwx	chmod 0777 filename; chmod -R 0777 dir	All classes can read/write/execute
-rwxr--r--	chmod 0744 filename; chmod -R 0744 dir	user can read/write/execute; all others can read
-rw-r--r--	chmod 0644 filename; chmod -R 0644 dir	user class can read/write; all others can read
-rw-rw-rw-	chmod 0666 filename' chmod -R 0666 dir	All classes can read/write

a look at how we can use the chmod command in our terminal to change permissions to anything we'd like!

These are just some examples. Using your new-found knowledge, you can set any permissions you'd like! Just be careful and make sure you don't break your system.

Access Control Lists(ACL) in Linux

What is ACL ?

Access control list (ACL) provides an additional, more flexible permission mechanism for file systems. It is designed to assist with UNIX file permissions. ACL allows you to give permissions for any user or group to any disc resource

Use of ACL :

Think of a scenario in which a particular user is not a member of group created by you but still you want to give some read or write access, how can you do it without making user a member of group, here comes in picture Access Control Lists, ACL helps us to do this trick.

Basically, ACLs are used to make a flexible permission mechanism in Linux.

setfacl and **getfacl** are used for setting up ACL and showing ACL respectively.

For example ,

```
getfacl test/seinfeld.txt
```

Output:

```
# file: test/seinfeld.txt
# owner: iafzal
# group: iafzal
user::rw-
group::rw-
other::r--
```

List of commands for setting up ACL :

1) To add permission for a user

```
setfacl -m "u:user:permissions" /path/to/file
```

2) To add permissions for a group

```
setfacl -m "g:group:permissions" /path/to/file
```

3) To allow all files or directories to inherit ACL entries from the directory it is within

```
setfacl -dm "entry" /path/to/dir
```

4) To remove a specific entry

```
setfacl -x "entry" /path/to/file
```

5) To remove all entries

```
setfacl -b path/to/file
```

For example :

```
setfacl -m u:iafzal:rwx test/seinfeld.txt
```

Modifying ACL using setfacl :

To add permissions for a user (user is either the user name or ID):

```
# setfacl -m "u:user:permissions"
```

To add permissions for a group (group is either the group name or ID):

```
# setfacl -m "g:group:permissions"
```

To allow all files or directories to inherit ACL entries from the directory it is within:

```
# setfacl -dm "entry"
```

Example :

```
setfacl -m u:iafzal:r-x test/seinfeld.txt
```

setfacl and getfacl

View ACL :

To show permissions :

```
# getfacl filename
```

Observe the difference between output of **getfacl** command before and after setting up ACL permissions using **setfacl** command.

Remove ACL :

If you want to remove the set ACL permissions, use **setfacl** command with **-b** option.

For example :

remove set permissions

If you compare output of getfacl command before and after using setfacl command with -b option, you can observe that there is no particular entry for user iafzal in later output.

You can also check if there are any extra permissions set through ACL using ls command.

check set acl with ls

Observe the first command output in image, there is extra "+" sign after the permissions like **-rw-rwrxr-** +, this indicates there are extra ACL permissions set which you can check by getfacl command

HANDOUTS

MODULE 5

WELCOME TO: MODULE 5

**LINUX SYSTEM
ADMINISTRATION**

Linux File Editor

- A text editor is a program which enables you to create and manipulate data (text) in a Linux file
- There are several standard text editors available on most Linux systems
 - **vi** – **Visual editor**
 - **ed** – **Standard line editor**
 - **ex** – **Extended line editor**
 - **emacs** – **A full screen editor**
 - **pico** – **Beginner's editor**
 - **vim** – **Advance version of vi**
- Our editor = vi (available in almost every Linux distribution)

Introduction to vi Editor

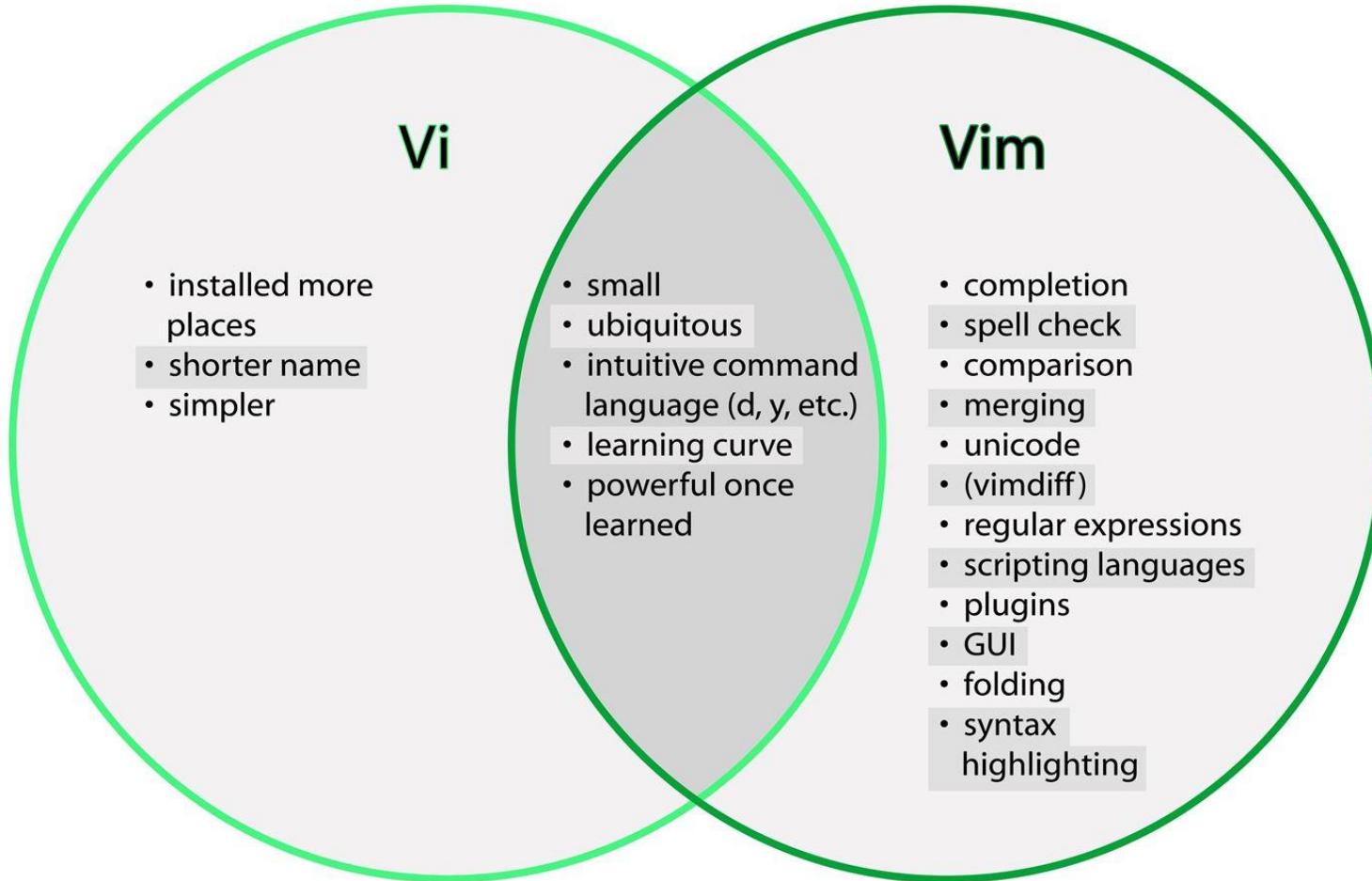
- vi supplies commands for:
 - Inserting and deleting text
 - Replacing text
 - Moving around the file
 - Finding and substituting strings
 - Cutting and pasting text
- Most common keys:
 - **i** - **insert**
 - **Esc** - **Escape out of any mode**
 - **r** - **replace**
 - **d** - **delete**
 - **:q!** - **quit without saving**
 - **:wq!** - **quit and save**

Difference Between vi and vim Editor

- As far as functionality is concerned, both editors work in the same manner. Which editor you choose is a matter of personal choice. Some people recommend learning the vim editor instead of the vi editor. Due to added features, learning and using vim editor is much easier than the vi editor.
- Since vim is based on the vi, when you will learn how to use the vim editor, you will automatically learn how to use the vi editor.
- vim has all the features as vi with some excellent addition
- There's also a comprehensive help system and lots of customization options available.



Difference Between vi and vim Editor



“vim” Interactive Learning Tools

- There are many websites that offer free vim interactive training:
 - <https://www.openvim.com/>
 - <http://www.vimgenius.com>
 - <https://vim-adventures.com/> (Games)



“sed” Command

- Replace a string in a file with a newstring
- Find and delete a line
- Remove empty lines
- Remove the first or n lines in a file
- To replace tabs with spaces
- Show defined lines from a file
- Substitute within vi editor
- And much more...

User Account Management

Commands

- `useradd`
- `groupadd`
- `userdel`
- `groupdel`
- `usermod`

Files

- `/etc/passwd`
- `/etc/group`
- `/etc/shadow`

Example:

```
useradd -g superheros -s /bin/bash -c "user description" -m -d  
/home/spiderman spiderman
```

The /etc/login.def File

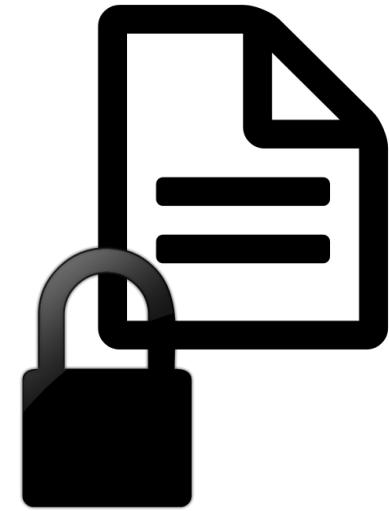
- The chage command – per user

- **Example**

```
chage [-m mindays] [-M maxdays] [-d lastday] [-I inactive] [-E  
expiredate] [-W warndays] user
```

- File = /etc/login.def

- PASS_MAX_DAYS 99999
 - PASS_MIN_DAYS 0
 - PASS_MIN_LEN 5
 - PASS_WARN_AGE 7



The chage Command

- The chage command – per user

- **Example**

```
chage [-d lastday] [-m mindays] [-M maxdays] [-W warndays] [-I
inactive] [-E expiredate] user
```



-d = 3. Last password change (lastchanged) : Days since Jan 1, 1970 that password was last changed

-m = 4. Minimum : The minimum number of days required between password changes i.e. the number of days left before the user is allowed to change his/her password

-M = 5. Maximum : The maximum number of days the password is valid (after that user is forced to change his/her password)

-W = 6. Warn : The number of days before password is to expire that user is warned that his/her password must be changed

-I = 7. Inactive : The number of days after password expires that account is disabled

-E = 8. Expire : days since Jan 1, 1970 that account is disabled i.e. an absolute date specifying when the login may no longer be used.

Switch Users and sudo Access

Commands

- su – username
- sudo command
- visudo

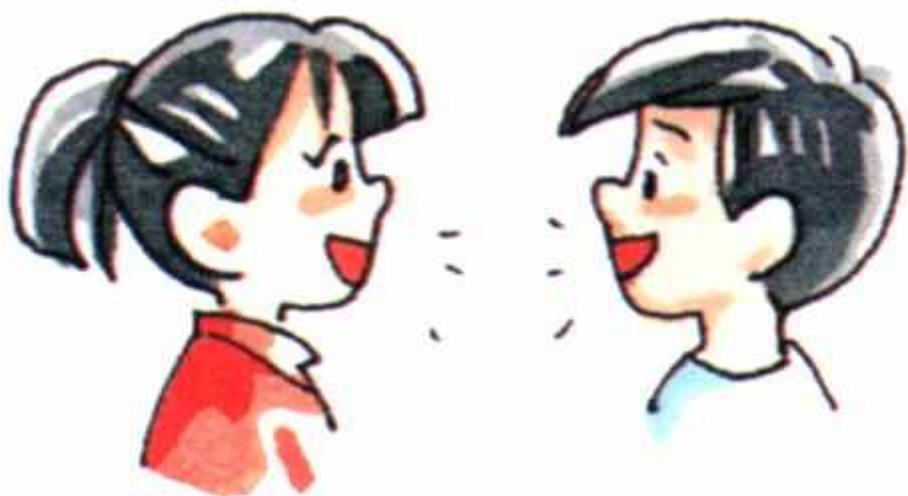
File

- /etc/sudoers

Monitor Users

- who
- last
- w
- finger
- id

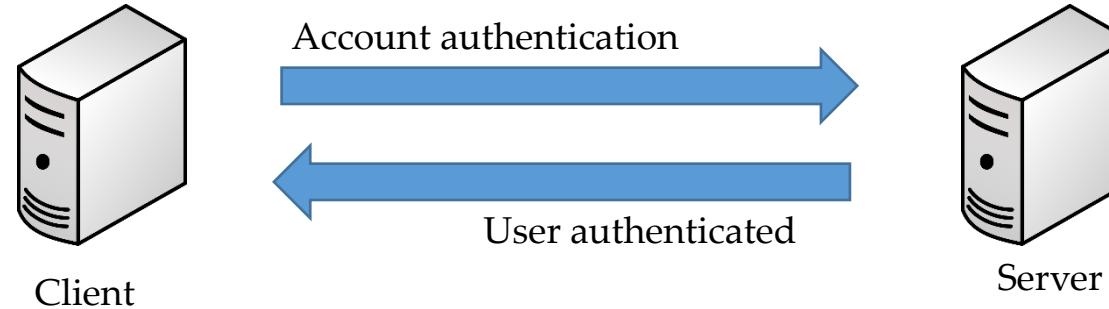
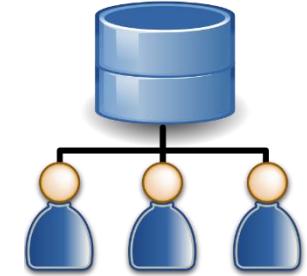
Talking to Users



- users
- wall
- write

Linux Account Authentication

- Types of Accounts
 - Local accounts
 - Domain/Directory accounts



- Windows = Active Directory
- Linux = LDAP?

Difference between Active Directory, LDAP, IDM, WinBIND, OpenLDAP etc.

- Active Directory = Microsoft
- IDM = Identity Manager
- WinBIND = Used in Linux to communicate with Windows (Samba)
- OpenLDAP (open source)
- IBM Directory Server
- JumpCloud
- LDAP = Lightweight Directory Access Protocol

System Utility Commands

- **date**
- **uptime**
- **hostname**
- **uname**
- **which**
- **cal**
- **bc**

Processes and Jobs

- Application = Service
- Script
- Process
- Daemon
- Threads
- Job

Process / Services Commands

- systemctl or service
- ps
- top
- kill
- crontab
- at.

systemctl command

- systemctl command is a new tool to control system services
- It is available in version 7 and later and it replaces the **service** command
- Usage example:

```
systemctl start|stop|status servicename.service      (firewalld)
```

```
systemctl enable servicename.service
```

```
systemctl restart|reload servicename.service
```

```
systemctl list-units --all
```

The output has the following columns:

- **UNIT**: The `systemd` unit name
- **LOAD**: Whether the unit's configuration has been parsed by `systemd`. The configuration of loaded units is kept in memory.
- **ACTIVE**: A summary state about whether the unit is active. This is usually a fairly basic way to tell if the unit has started successfully or not.
- **SUB**: This is a lower-level state that indicates more detailed information about the unit. This often varies by unit type, state, and the actual method in which the unit runs.
- **DESCRIPTION**: A short textual description of what the unit is/does.

systemctl command

- To add a service under systemctl management:

Create a unit file in **/etc/systemd/system/servicename.service**

- To control system with systemctl

systemctl poweroff

systemctl halt

systemctl reboot

“ps” command

- **ps** command stands for process status and it displays all the currently running processes in the Linux system

Usage examples:

- **ps** = Shows the processes of the current shell

PID = the unique process ID

TTY = terminal type that the user logged-in to

TIME = amount of CPU in minutes and seconds that the process has been running

CMD = name of the command

- **ps -e** = Shows all running processes
- **ps aux** = Shows all running processes in BSD format
- **ps -ef** = Shows all running processes in full format listing (*Most commonly used*)
- **ps -u username** = Shows all processes by username.

“top” command

- top command is used to show the Linux processes and it provides a real-time view of the running system
- This command shows the summary information of the system and the list of processes or threads which are currently managed by the Linux Kernel
- When the top command is executed then it goes into interactive mode and you can exit out by hitting **q**
- **Usage:** **top**

PID: Shows task's unique process id

USER: Username of owner of task

PR: The “PR” field shows the scheduling priority of the process from the perspective of the kernel

NI: Represents a Nice Value of task. A Negative nice value implies higher priority, and positive Nice value means lower priority.

VIRT: Total virtual memory used by the task

RES: Memory consumed by the process in RAM

SHR: Represents the amount of shared memory used by a task

S: This field shows the process state in the single-letter form

%CPU: Represents the CPU usage

%MEM: Shows the Memory usage of task

TIME+: CPU Time, the same as ‘TIME’, but reflecting more granularity through hundredths of a second.

“top” command

- **top -u iafzal** = shows tasks/processes by user owned
- **top then press c** = shows commands absolute path
- **top then press k** = kill a process by PID within top session
- **top then M and P** = To sort all Linux running processes by Memory usage

Please note:

Top command refreshes the information every 3 seconds

“kill” command

- **kill** command is used to terminate processes manually
- It sends a signal which ultimately terminates or kills a particular process or group of processes

Usage:

kill [OPTION] [PID]

OPTION = Signal name or signal number/ID

PID = Process ID

kill -1 = to get a list of all signal names or signal number

Most used signals are:

kill PID = Kill a process with default signal

kill -1 = Restart

kill -2 = Interrupt from the keyboard just like Ctrl C

kill -9 = Forcefully kill the process

kill -15 = Kill a process gracefully

“kill” command

- Other similar kill commands are:

killall

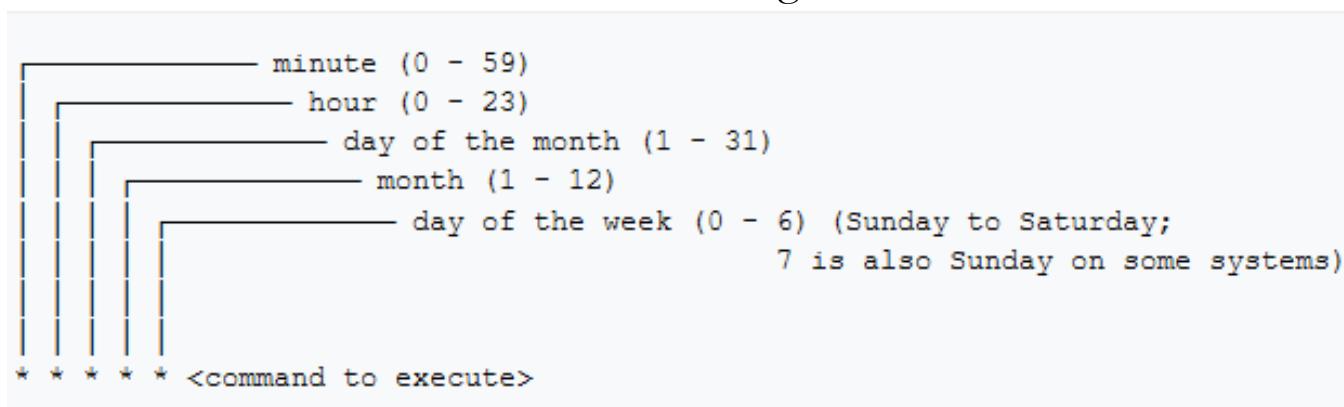
pkill

“crontab” command

- Crontab command is used to schedule tasks

Usage:

- **crontab -e** = Edit the crontab
- **crontab -l** = List the crontab entries
- **crontab -r** = Remove the crontab
- **crond** = crontab daemon/service that manages scheduling
- **systemctl status crond** = To manage the crond service



- Create crontab entry by scheduling a task:

```
crontab -e  
schedule, echo "This is my first crontab entry" > crontab-entry
```

“at” command

- at command is like crontab which allows you to schedule jobs but only once
- When the command is run it will enter interactive mode and you can get out by pressing **Ctrl D**

Usage:

- **at HH:MM PM** = Schedule a job
- **atq** = List the at entries
- **atrm #** = Remove at entry
- **atd** = at daemon/service that manages scheduling
- **systemctl status atd** = To manage the atd service

- Create at entry by scheduling a task:

```
at 4:45PM → enter  
echo "This is my first at entry" > at-entry  
Crtl D
```

“at” command

Other future scheduling format:

- **at 2:45 AM 101621** = Schedule a job to run on Oct 16th, 2021 at 2:45am
- **at 4PM + 4 days** = Schedule a job at 4pm four days from now
- **at now +5 hours** = Schedule a job to run five hours from now
- **at 8:00 AM Sun** = Schedule a job to 8am on coming Sunday
- **at 10:00 AM next month** = Schedule a job to 10am next month

Additional Cron Jobs

- By default, there are 4 different types of cronjobs
 - Hourly
 - Daily
 - Weekly
 - Monthly
- All the above crons are setup in
 - **/etc/cron.____** (directory)
- The timing for each are set in
 - **/etc/anacrontab** -- except hourly
- For hourly
 - **/etc/cron.d/0hourly**



Process Management

- Background = Ctrl-z, jobs and bg
- Foreground = fg
- Run process even after exit = nohup process &
OR = nohup process > /dev/null 2>&1 &
- Kill a process by name = pkill
- Process priority = nice (e.g. nice -n 5 process)
The niceness scale goes from -20 to 19. The lower the number more priority that task gets
- Process monitoring = top
- List process = ps.

System Monitoring

- **top**
- **df**
- **dmesg**
- **iostat 1**
- **netstat**
- **free**
- **cat /proc/cpuinfo**
- **cat /proc/meminfo**

Log Monitoring

Another and most important way of system administration is log monitor

Log Directory = **/var/log**

- **boot**
- **chronyd** = NTP
- **cron**
- **maillog**
- **secure**
- **messages**
- **httpd**

System Maintenance Commands

- shutdown
- init 0-7
- reboot
- halt

Changing System Hostname

- **hostnamectl - set-hostname newhostname**
- **Version 7 = Edit /etc/hostname**
- **Version 6 = Edit /etc/sysconfig/network**

Finding System Information

- `cat /etc/redhat-release`
- `uname -a`
- `dmldecode`

System Architecture

- Differences between a 32-bit and 64-bit CPU

A big difference between 32-bit processors and 64-bit processors is the number of calculations per second they can perform, which affects the speed at which they can complete tasks. 64-bit processors can come in dual core, quad core, six core, and eight core versions for home computing. Multiple cores allow for an increased number of calculations per second that can be performed, which can increase the processing power and help make a computer run faster. Software programs that require many calculations to function smoothly can operate faster and more efficiently on the multi-core 64-bit processors

- Linux = arch
- Windows = My computer → Properties

Terminal Control Keys

Several key combinations on your keyboard usually have a special effect on the terminal.

These "control" (CTRL) keys are accomplished by holding the CTRL key while typing the second key.

For example, CTRL-c means to hold the CTRL key while you type the letter "c".

The most common control keys are listed below:

- **CTRL-u** - **erase everything you've typed on the command line**
- **CTRL-c** - **stop/kill a command**
- **CTRL-z** - **suspend a command**
- **CTRL-d** - **exit from an interactive program (signals end of data).**

Terminal Commands

- **clear**

Clears your screen

- **exit**

Exit out of the shell, terminal or a user session

- **script**

The script command stores terminal activities in a log file that can be named by a user, when a name is not provided by a user, the default file name, typescript is used

Recover Root Password

- Restart your computer
- Edit grub
- Change password
- reboot

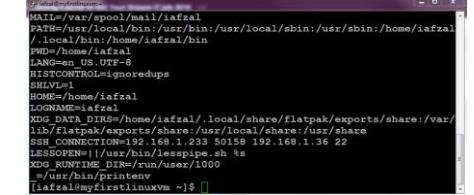
SOS Report



- What is SOS Report?
 - Collect and package diagnostic and support data
- Package name
 - sos-version
- Command
 - sosreport

Environment Variables

- What are environment variables?
 - An environment variable is a dynamic-named value that can affect the way running processes will behave on a computer. They are part of the environment in which a process runs.
 - In simple words: set of defined rules and values to build an environment
 - E.g.

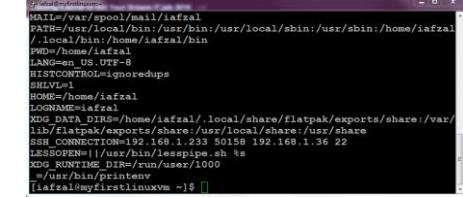


```
MAIL=/var/spool/mail/iafzal
PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/iafzal
./local/bin:/home/iafzal/bin
PWD=/home/iafzal
LANG=en_US.UTF-8
HISTCONTROL=ignore_dups
SHLVL=1
HOME=/home/iafzal
LOGNAME=iafzal
XDG_DATA_DIRS=/home/iafzal/.local/share/flatpak/exports/share:/var/lib/flatpak/exports/share:/usr/local/share:/usr/share
SSH_CONNECTION=192.168.1.233 50158 192.168.1.36 22
LESSOPEN=||/usr/bin/lesspipe.sh %s
XDG_RUNTIME_DIR=/run/user/1000
=/usr/bin/printenv
[iafzal@myfirstlinuxvm ~]$
```



Environment Variables

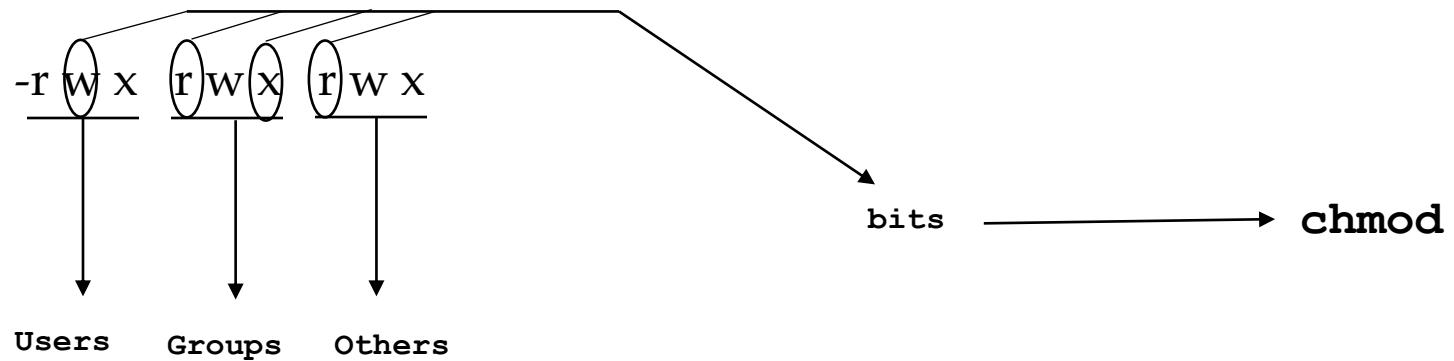
- To view all environment variables
 - **printenv OR env**
- To view ONE environment variable
 - **echo \$SHELL**
- To set the environment variables
 - **export TEST=1**
 - **echo \$TEST**
- To set environment variable permanently
 - **vi .bashrc**
 - **TEST='123'**
 - **export TEST**
- To set global environment variable permanently
 - **vi /etc/profile or /etc/bashrc**
 - **Test='123'**
 - **export TEST**



```
MAIL=/var/spool/mail/iafzal
PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/iafzal
./local/bin:/home/iafzal/bin
PWD=/home/iafzal
LANG=en_US.UTF-8
HISTCONTROL=ignore_dups
SHLVL=1
HOME=/home/iafzal
LOGNAME=iafzal
XDG_DATA_DIRS=/home/iafzal/.local/share/flatpak/exports/share:/var/
.flatpak/exports/share:/usr/local/share:/usr/share
SSH_CONNECTION=192.168.1.233 50158 192.168.1.36 22
LESSOPEN=||/usr/bin/lesspipe.sh %s
XDG_RUNTIME_DIR=/run/user/1000
=/usr/bin/printenv
[iafzal@myfirstlinuxvm ~]$
```

Special Permissions with `setuid`, `setgid` and `sticky bit`

- All permissions on a file or directory are referred as bits



- There are 3 additional permissions in Linux
 - **setuid**: bit tells Linux to run a program with the effective user id of the owner instead of the executor: (e.g. `passwd` command) → `/etc/shadow`
 - **setgid**: bit tells Linux to run a program with the effective group id of the owner instead of the executor: (e.g. `locate` or `wall` command)
 - **sticky bit**: a bit set on files/directories that allows only the owner or root to delete those files
- Please note: This bit is present for only files which have executable permissions*

Not actual commands

Special Permissions with `setuid`, `setgid` and sticky bit

- To assign special permissions at the user level
 - `chmod u+s xyz.sh`
- To assign special permissions at the group level
 - `chmod g+s xyz.sh`
- To remove special permissions at the user or group level
 - `chmod u-s xyz.sh`
 - `chmod g-s xyz.sh`
- To find all executables in Linux with setuid and setgid permissions
 - `find / -perm /6000 -type f`

Please note:

*These bits work on c
programming executables not
on bash shell scripts*

Sticky bit

- It is assigned to the last bit of permissions

`-r w x r w x r w(t)`

- Why? Example of `/tmp` directory

Special Permissions with `setuid`, `setgid` and `sticky bit`

Lab exercise:

- Become root and create a directory allinone in / = `mkdir /allinone`
- Assign all rwx permissions to that directory = `chmod 777 /allinone`
- Become iafzal and create directory inside of /allinone = `mkdir imrandir`
- Give all rwx permissions to that directory = `chmod 777 imrandir`
- Create 3 files in that directory = `touch a b c`
- Open another terminal and login as spiderman
- Go to /allinone directory and delete imrandir directory = `rm -rf imrandir`
 - *You will see the directory is deleted*
- Now become root again and assign sticky bit permission to /allinone = `chmod +t /allinone`
- Become iafzal and create directory again inside of /allinone = `mkdir imrandir`
- Give all rwx permissions to that directory = `chmod 777 imrandir`
- Create 3 files in that directory = `touch a b c`
- Become spiderman user again
- Go to /allinone directory and try to delete imrandir directory = `rm -rf imrandir`
 - *Now as spiderman you cannot delete the directory*

vi Commands

Entering vi

- vi filename - The filename can be the name of an existing file or the name of the file you want to create.
- view filename - Starts vi in "read only" mode. Allows you to look at a file without the risk of altering its contents.

Exiting vi

- :q - quit - if you have made any changes, vi will warn you of this, and you'll need to use one of the other quits.
- :w - write edit buffer to disk
- :w filename - write edit buffer to disk as filename
- :wq - write edit buffer to disk and quit
- ZZ - write edit buffer to disk and quit
- :q! - quit without writing edit buffer to disk

Positioning within text

By character

- left arrow - left one character
- right arrow - right one character
- backspace - left one character
- space - right one character
- h - left one character
- l - right one character

By word

- w - beginning of next word
- nw - beginning of nth next word
- b - back to previous word
- nb - back to nth previous word
- e - end of next word
- ne - end of nth next word

By line

down arrow	- down one line
up arrow	- up one line
j	- down one line
k	- up one line
+	- beginning of next line down
-	- beginning of previous line up
0	- first column of current line (zero)
^	- first character of current line
\$	- last character of current line

By block

(- beginning of sentence
)	- end of sentence
{	- beginning of paragraph
}	- end of paragraph

By screen

CTRL-f	- forward 1 screen
CTRL-b	- backward 1 screen
CTRL-d	- down 1/2 screen
CTRL-u	- up 1/2 screen
H	- top line on screen
M	- mid-screen
L	- last line on screen

Within file

nG	- line n within file
1G	- first line in file
G	- last line in file

Begin the vi editor exercises

Inserting text

a	- append text after cursor *
A	- append text at end of line *
i	- insert text before cursor *
I	- insert text at beginning of line *
o	- open a blank line after the current line for text input *
O	- open a blank line before the current

line for text input *

* Note: hit ESC (escape) key when finished inserting!

Continue the vi exercises

Deleting text

x	- delete character at cursor
dh	- delete character before cursor
nx	- delete n characters at cursor
dw	- delete next word
db	- delete previous word
dnw	- delete n words from cursor
dnb	- delete n words before cursor
d0	- delete to beginning of line
d\$	- delete to end of line
D	- delete to end of line
dd	- delete current line
d(- delete to beginning of sentence
d)	- delete to end of sentence
d{	- delete to beginning of paragraph
d}	- delete to end of paragraph
ndd	- delete n lines (start at current line)

Changing text

cw	- replace word with text *
cc	- replace line with text *
c0	- change to beginning of line *
c\$	- change to end of line *
C	- change to end of line *
c(- change to beginning of sentence *
c)	- change to end of sentence *
c{	- change to beginning of paragraph *
c}	- change to end of paragraph *
r	- overtype only 1 character
R	- overtype text until ESC is hit *
J	- join two lines

* Note: hit ESC (escape) key when finished changing!

Copying lines

yy	- "yank": copy 1 line into buffer
nyy	- "yank": copy n lines into buffer
p	- put contents of buffer after current line
P	- put contents of buffer before current line

Moving lines (cutting and pasting)

ndd	- delete n lines (placed in buffer)
p	- put contents of buffer after current line
P	- put contents of buffer before current line

Searching / Substituting

/str	- search forward for str
?str	- search backward for str
n	- find next occurrence of current string
N	- repeat previous search in reverse direction

The substitution command requires a line range specification. If it is omitted, the default is the current line only. The examples below show how to specify line ranges.

:s/old/new	- substitute new for first occurrence of old in current line
:s/old/new/g	- substitute new for all occurrences of old in current line
:1,10s/old/new	- substitute new for first occurrence of old in lines 1 - 10
:.,\$s/old/new	- substitute new for first occurrence of old in remainder of file
:.,+5s/old/new	- substitute new for first occurrence of old in current line and next 5 lines
:.,-5s/old/new	- substitute new for first occurrence of old in current line and previous 5 lines
:%s/old/new/g	- substitute new for all occurrences of old in the entire file

```
:%s/old/new/gc - interactively substitute new for all  
occurrences of old - will prompt for  
y/n response for each substitution.
```

Miscellaneous commands

u	- undo the last command (including undo)
.	- repeat last command
xp	- swap two adjacent characters
m[a-z]	- set a marker (a - z)
'[a-z]	- go to a previously set marker (a - z)
:!command	- execute specified LINUX command
:r filename	- read/insert contents of filename after current line.
:1,100!fmt	- reformat the first 100 lines
:!fmt	- reformat the entire file

vi Options

You can change the way vi operates by changing the value of certain options which control specific parts of the vi environment.

To set an option during a vi session, use one of the commands below as required by the option:

```
:set option_name  
:set option_name=value
```

Some examples of the more common options are described below.

:set all	- shows all vi options in effect
:set ai	- set autoindent - automatically indents each line of text
:set noai	- turn autoindent off
:set nu	- set line numbering on

```
:set nonu      - turn line numbering off

:set scroll=n  - sets number of lines to be scrolled
                to n. Used by screen scroll commands.

:set sw=n      - set shiftwidth to n. Used by autoindent
                option.

:set wm=n      - set wrapmargin to n. Specifies number
                of spaces to leave on right edge of the
                screen before wrapping words to next
                line.

:set showmode   - reminds you when you are inserting
                text.

:set ic        - ignore case of characters when
                performing a search.
```

Options can be set permanently by putting them in a file called .exrc in your home directory. A sample .exrc file appears below. Note that you do not need the colon (:) as part of the option specification when you put the commands in a .exrc file. Also note that you can put them all on one line.

```
set nu ai wm=5 showmode ic
```

User Account Management:

Following are the basic user account management commands

- **useradd**
To create a new user in Linux. A different options can be used to modify userId, home directory etc.
- **userdel**
This command is used to delete the user. Please note this command alone will not delete the user home directory. You will have to use option –r to delete user home directory
- **groupadd**
Creates a new group
- **groupdel**
Removes an existing group
- **usermod**
Modify user attributes such as user home directory, user group, user ID etc.

User Files

- /etc/passwd = This file has all user's attributes
- /etc/shadow = This file contains encrypted user password and password policy
- /etc/group = All group and user group information

Creating User Accounts in Linux:

When we run ‘useradd’ command in Linux terminal, it performs following major things:

It edits /etc/passwd, /etc/shadow, /etc/group and /etc/gshadow files for the newly created User account.
Creates and populate a home directory for the new user.
Sets permissions and ownerships to home directory.

Basic syntax of command is:

```
useradd [options] username
```

In this article we will show you the most used 15 useradd commands with their practical examples in Linux. We have divided the section into two parts from Basic to Advance usage of command.

Part I: Basic usage with 10 examples

Part II: Advance usage with 5 examples

Part I – 10 Basic Usage of useradd Commands

1. How to Add a New User in Linux

To add/create a new user, all you’ve to follow the command ‘useradd’ or ‘adduser’ with ‘username’. The ‘username’ is a user login name, that is used by user to login into the system.

Only one user can be added and that username must be unique (different from other username already exists on the system).

For example, to add a new user called ‘solider’, use the following command.

```
[root@localhost ~]# useradd solider
```

When we add a new user in Linux with ‘useradd’ command it gets created in locked state and to unlock that user account, we need to set a password for that account with ‘passwd’ command.

```
[root@localhost ~]# passwd solider
Changing password for user solider.
New LINUX password:
Retype new LINUX password:
passwd: all authentication tokens updated successfully.
```

Once a new user created, it's entry automatically added to the '/etc/passwd' file. The file is used to store users information and the entry should be.

```
solider:x:504:504:solider:/home/solider:/bin/bash
```

The above entry contains a set of seven colon-separated fields, each field has it's own meaning. Let's see what are these fields:

Username: User login name used to login into system. It should be between 1 to 32 characters long.

Password: User password (or x character) stored in /etc/shadow file in encrypted format.

User ID (UID): Every user must have a User ID (UID) User Identification Number. By default UID 0 is reserved for root user and UID's ranging from 1-99 are reserved for other predefined accounts. Further UID's ranging from 100-999 are reserved for system accounts and groups.

Group ID (GID): The primary Group ID (GID) Group Identification Number stored in /etc/group file.

User Info: This field is optional and allow you to define extra information about the user. For example, user full name. This field is filled by 'finger' command.

Home Directory: The absolute location of user's home directory.

Shell: The absolute location of a user's shell i.e. /bin/bash.

2. Create a User with Different Home Directory

By default 'useradd' command creates a user's home directory under /home directory with username. Thus, for example, we've seen above the default home directory for the user 'solider' is '/home/solider'.

However, this action can be changed by using '-d' option along with the location of new home directory (i.e. /home/newusers). For example, the following command will create a user 'solider' with a home directory '/home/newusers'.

```
[root@localhost ~]# useradd -d /home/newusers solider
```

You can see the user home directory and other user related information like user id, group id, shell and comments.

```
[root@localhost ~]# cat /etc/passwd | grep solider
solider:x:505:505::/home/newusers:/bin/bash
```

3. Create a User with Specific User ID

In Linux, every user has its own UID (Unique Identification Number). By default, whenever we create a new user accounts in Linux, it assigns userid 500, 501, 502 and so on...

But, we can create user's with custom userid with '-u' option. For example, the following command will create a user 'navin' with custom userid '999'.

```
[root@localhost ~]# useradd -u 999 navin
```

Now, let's verify that the user created with a defined userid (999) using following command.

```
[root@localhost ~]# cat /etc/passwd | grep solider  
navin:x:999:999::/home/solider:/bin/bash
```

NOTE: Make sure the value of a user ID must be unique from any other already created users on the system.

4. Create a User with Specific Group ID

Similarly, every user has its own GID (Group Identification Number). We can create users with specific group ID's as well with -g option.

Here in this example, we will add a user 'tarunika' with a specific UID and GID simultaneously with the help of '-u' and '-g' options.

```
[root@localhost ~]# useradd -u 1000 -g 500 tarunika
```

Now, see the assigned user id and group id in '/etc/passwd' file.

```
[root@localhost ~]# cat /etc/passwd | grep tarunika  
tarunika:x:1000:500::/home/tarunika:/bin/bash
```

5. Add a User to Multiple Groups

The '-G' option is used to add a user to additional groups. Each group name is separated by a comma, with no intervening spaces.

Here in this example, we are adding a user 'solider' into multiple groups like admins, webadmin and developer.

```
[root@localhost ~]# useradd -G admins,webadmin,developers solider
```

Next, verify that the multiple groups assigned to the user with id command.

```
[root@localhost ~]# id solider  
uid=1001(solider) gid=1001(solider)  
groups=1001(solider),500(admins),501(webadmin),502(developers)  
context=root:system_r:unconfined_t:SystemLow-SystemHigh
```

6. Add a User without Home Directory

In some situations, where we don't want to assign a home directories for a user's, due to some security reasons. In such situation, when a user logs into a system that has just restarted, its home directory will be root. When such user uses su command, its login directory will be the previous user home directory.

To create user's without their home directories, '-M' is used. For example, the following command will create a user 'shilpi' without a home directory.

```
[root@localhost ~]# useradd -M shilpi
```

Now, let's verify that the user is created without home directory, using ls command.

```
[root@localhost ~]# ls -l /home/shilpi  
ls: cannot access /home/shilpi: No such file or directory
```

7. Create a User with Account Expiry Date

By default, when we add user's with 'useradd' command user account never get expires i.e their expiry date is set to 0 (means never expired).

However, we can set the expiry date using '-e' option, that sets date in YYYY-MM-DD format. This is helpful for creating temporary accounts for a specific period of time.

Here in this example, we create a user 'aparna' with account expiry date i.e. 27th April 2014 in YYYY-MM-DD format.

```
[root@localhost ~]# useradd -e 2014-03-27 aparna
```

Next, verify the age of account and password with 'chage' command for user 'aparna' after setting account expiry date.

```
[root@localhost ~]# chage -l aparna  
Last password change : Mar 28, 2014  
Password expires : never  
Password inactive : never  
Account expires : Mar 27, 2014  
Minimum number of days between password change : 0  
Maximum number of days between password change : 99999  
Number of days of warning before password expires : 7
```

8. Create a User with Password Expiry Date

The '-f' argument is used to define the number of days after a password expires. A value of 0 inactive the user account as soon as the password has expired. By default, the password expiry value set to -1 means never expire.

Here in this example, we will set a account password expiry date i.e. 45 days on a user 'solider' using '-e' and '-f' options.

```
[root@localhost ~]# useradd -e 2014-04-27 -f 45 solider
```

9. Add a User with Custom Comments

The '-c' option allows you to add custom comments, such as user's full name, phone number, etc to /etc/passwd file. The comment can be added as a single line without any spaces.

For example, the following command will add a user 'mansi' and would insert that user's full name, Manis Khurana, into the comment field.

```
[root@localhost ~]# useradd -c "Manis Khurana" mansi
```

You can see your comments in '/etc/passwd' file in comments section.

```
[root@localhost ~]# tail -1 /etc/passwd  
mansi:x:1006:1008:Manis Khurana:/home/mansi:/bin/sh
```

10. Change User Login Shell:

Sometimes, we add users which has nothing to do with login shell or sometimes we require to assign different shells to our users. We can assign different login shells to a each user with '-s' option.

Here in this example, will add a user 'solider' without login shell i.e. '/sbin/nologin' shell.

```
[root@localhost ~]# useradd -s /sbin/nologin solider
```

You can check assigned shell to the user in '/etc/passwd' file.

```
[root@localhost ~]# tail -1 /etc/passwd  
solider:x:1002:1002::/home/solider:/sbin/nologin
```

Part II – 5 Advance Usage of useradd Commands

11. Add a User with Specific Home Directory, Default Shell and Custom Comment

The following command will create a user 'ravi' with home directory '/var/www/solider', default shell /bin/bash and adds extra information about user.

```
[root@localhost ~]# useradd -m -d /var/www/ravi -s /bin/bash -c "Solider Owner" -U ravi
```

In the above command ‘-m -d’ option creates a user with specified home directory and the ‘-s’ option set the user’s default shell i.e. /bin/bash. The ‘-c’ option adds the extra information about user and ‘-U’ argument create/adds a group with the same name as the user.

12. Add a User with Home Directory, Custom Shell, Custom Comment and UID/GID

The command is very similar to above, but here we defining shell as ‘/bin/zsh’ and custom UID and GID to a user ‘tarunika’. Where ‘-u’ defines new user’s UID (i.e. 1000) and whereas ‘-g’ defines GID (i.e. 1000).

```
[root@localhost ~]# useradd -m -d /var/www/tarunika -s /bin/zsh -c "Solider Technical Writer" -u 1000 -g 1000 tarunika
```

13. Add a User with Home Directory, No Shell, Custom Comment and User ID

The following command is very much similar to above two commands, the only difference is here, that we disabling login shell to a user called ‘avishek’ with custom User ID (i.e. 1019).

Here ‘-s’ option adds the default shell /bin/bash, but in this case we set login to ‘/usr/sbin/nologin’. That means user ‘avishek’ will not able to login into the system.

```
[root@localhost ~]# useradd -m -d /var/www/avishek -s /usr/sbin/nologin -c "Solider Sr. Technical Writer" -u 1019 avishek
```

14. Add a User with Home Directory, Shell, Custom Skell/Comment and User ID

The only change in this command is, we used ‘-k’ option to set custom skeleton directory i.e. /etc/custom.skell, not the default one /etc/skel. We also used ‘-s’ option to define different shell i.e. /bin/tcsh to user ‘navin’.

```
[root@localhost ~]# useradd -m -d /var/www/navin -k /etc/custom.skell -s /bin/tcsh -c "No Active Member of Solider" -u 1027 navin
```

15. Add a User without Home Directory, No Shell, No Group and Custom Comment

This following command is very different than the other commands explained above. Here we used ‘-M’ option to create user without user’s home directory and ‘-N’ argument is used that tells the system to only create username (without group). The ‘-r’ arguments is for creating a system user.

```
[root@localhost ~]# useradd -M -N -r -s /bin/false -c "Disabled Solider Member" clayton
```

For more information and options about useradd, run ‘useradd’ command on the terminal to see available options.

Read Also: 15 usermod Command Examples

Share

+

0

0

0

Ask Anything

If You Appreciate What We Do Here On Solider, You Should Consider:

Stay Connected to: Twitter | Facebook | Google Plus

Subscribe to our email updates: Sign Up Now

Get your own self-hosted blog with a Free Domain at (\$3.95/month).

Become a Supporter - Make a contribution via PayPal

Support us by purchasing our premium books in PDF format.

Support us by taking our online Linux courses

We are thankful for your never ending support.

Tags: adduserlinux usersuseradd

[View all Posts](#)

Ravi Saive

I am Ravi Saive, creator of Solider. A Computer Geek and Linux Guru who loves to share tricks and tips on Internet. Most Of My Servers runs on Open Source Platform called Linux. Follow Me: Twitter, Facebook and Google+

Your name can also be listed here. Got a tip? Submit it here to become an Solider author.

[RedHat RHCE and RHCSA Certification Book](#)

[Linux Foundation LFCS and LFCE Certification Preparation Guide](#)

[Next story](#)

[Fun in Linux Terminal – Play with Word and Character Counts](#)

[Previous story](#)

[nSnake: A Clone of Old Classic Snake Game – Play in Linux Terminal](#)

You may also like...

[Find Number of Files in a Directory and Subdirectories](#)

0

[How to Find Number of Files in a Directory and Subdirectories](#)

17 Jan, 2017

Display Command File Contents in Column Format

0

Display Command Output or File Contents in Column Format

6 Feb, 2018

Convert RPM to DEB and DEB to RPM

8

How to Convert From RPM to DEB and DEB to RPM Package Using Alien

26 Aug, 2015

88 Responses

Comments4

Pingbacks0

Decontee K Sawyer

October 26, 2017 at 6:29 pm

Hi Ravi. Your suggestion to go directly to the source documentation to understand the requirements and details is an exceedingly excellent one. You have obviously done so, and translated the English it is written in, into whatever your native language is. A link to your interpretation, in your native language would be more helpful than the confusing broken English found here.

Reply

Anuj

October 16, 2017 at 4:01 pm

Hi Ravi,

I have one problem, from client side I have a request to add a new user with username having space, I mean username of two words.

For example,

```
# adduser "ravi gen"  
adduser: invalid user name 'ravi gen'
```

Reply

Ravi Saive

October 25, 2017 at 11:52 am

@Anuj,

That's not possible, add underscore or dash, like ravi_gen or ravi-gen.

[Reply](#)

oscar javier guerrero

October 6, 2017 at 9:19 pm

Hi Ravi, I have a question, If use: su "user" type the password and the system say: su: System Error, why is this message?

[Reply](#)

[« Older Comments](#)

Got something to say? Join the discussion.

[Comment](#)

Name *

Email *

Website

Notify me of followup comments via e-mail. You can also subscribe without commenting.

Switch Users and Sudo Access:

Switch Users:

Following is the user switch command that can be used to switch from one user to another

- `su - username`
`su -` invokes a login shell after switching the user. A login shell resets most environment variables, providing a clean base.
- `su username`
just switches the user, providing a normal shell with an environment nearly the same as with the old user

Sudo Access:

- `sudo command-name`
The above command “`sudo command-name`” will run any command owned and authorized by root account as long as that user is authorized to run it in `/etc/sudoers` file

Configuring sudo Access

1. Log in to the system as the `root` user.
2. Create a normal user account using the `useradd` command. Replace `USERNAME` with the user name that you wish to create.

```
# useradd USERNAME
```

3. Set a password for the new user using the `passwd` command.
4. # `passwd USERNAME`
5. Changing password for user `USERNAME`.
6. New password:
7. Retype new password:
`passwd: all authentication tokens updated successfully.`

8. Run the `visudo` to edit the `/etc/sudoers` file. This file defines the policies applied by the `sudo` command.

```
# visudo
```

9. Find the lines in the file that grant `sudo` access to users in the group `wheel` when enabled.
10. ## Allows people in group wheel to run all commands
%wheel ALL=(ALL) ALL
11. Remove the comment character (#) at the start of the second line. This enables the configuration option.
12. Save your changes and exit the editor.
13. Add the user you created to the `wheel` group using the `usermod` command.

```
# usermod -aG wheel USERNAME
```

14. Test that the updated configuration allows the user you created to run commands using sudo.

1. Use the su to switch to the new user account that you created.

```
# su USERNAME -
```

2. Use the groups to verify that the user is in the wheel group.

```
3. $ groups  
USERNAME wheel
```

4. Use the sudo command to run the whoami command. As this is the first time you have run a command using sudo from this user account the banner message will be displayed. You will be also be prompted to enter the password for the user account.

```
5. $ sudo whoami  
6. We trust you have received the usual lecture from the local  
System  
7. Administrator. It usually boils down to these three things:  
8.  
9.      #1) Respect the privacy of others.  
10.     #2) Think before you type.  
11.     #3) With great power comes great responsibility.  
12.  
13. [sudo] password for USERNAME:  
root
```

The last line of the output is the user name returned by the whoami command. If sudo is configured correctly this value will be root.

You have successfully configured a user with sudo access. You can now log in to this user account and use sudo to run commands as if you were logged in to the account of the root user.

Linux Editors

- What is a text editor?
 - A text editor is a program which enables you to create and manipulate character data (text) in a computer file.
 - A text editor is not a word processor although some text editors do include word processing facilities.
 - Text editors often require "memorizing" commands in order to perform editing tasks. The more you use them, the easier it becomes. There is a "learning curve" in most cases though.
- There are several standard text editors available on most LINUX systems:
 - ed - standard line editor
 - ex - extended line editor
 - vi - a visual editor; full screen; uses ed/ex line-mode commands for global file editing
 - sed - stream editor for batch processing of files
- In addition to these, other local "favorites" may be available:
 - emacs - a full screen editor and much more
 - pico - an easy "beginner's" editor
 - lots of others

The Standard Display Editor - vi

- vi supplies commands for:
 - inserting and deleting text
 - replacing text
 - moving around the file
 - finding and substituting strings
 - cutting and pasting text
 - reading and writing to other files
- vi uses a "buffer"
 - While using vi to edit an existing file, you are actually working on a copy of the file that is held in a temporary buffer in your computer's memory.
 - If you invoked vi with a new filename, (or no file name) the contents of the file only exist in this buffer.
 - Saving a file writes the contents of this buffer to a disk file, replacing its contents. You can write the buffer to a new file or to some other file.
 - You can also decide not to write the contents of the buffer, and leave your original file unchanged.
- vi operates in two different "modes":
 - Command mode

- vi starts up in this mode
 - Whatever you type is interpreted as a command - not text to be inserted into the file.
 - The mode you need to be in if you want to "move around" the file.
- Insert mode
 - This is the mode you use to type (insert) text.
 - There are several commands that you can use to enter this mode.
 - Once in this mode, whatever you type is interpreted as text to be included in the file. You cannot "move around" the file in this mode.
 - Must press the ESC (escape) key to exit this mode and return to command mode.

Monitor User Commands:

Following are the basic user monitor commands

- who
- last
- w
- id

who

As a Linux user, sometimes it is required to know some basic information like :

- Time of last system boot
- List of users logged-in
- Current run level etc

Though this type of information can be obtained from various files in the Linux system but there is a command line utility 'who' that does exactly the same for you. In this article, we will discuss the capabilities and features provided by the 'who' command.

The basic syntax of the who command is :

```
who [OPTION]... [ FILE | ARG1 ARG2 ]
```

Examples of 'who' command

1. Get the information on currently logged in users

This is done by simply running the 'who' command (without any options). Consider the following example:

```
$ who
iafzal tty7      2012-08-07 05:33 (:0)
iafzal pts/0      2012-08-07 06:47 (:0.0)
iafzal pts/1      2012-08-07 07:58 (:0.0)
```

2. Get the time of last system boot

This is done using the -b option. Consider the following example:

```
$ who -b
                  system boot 2012-08-07 05:32
```

So we see that the above output gives the exact date and time of last system boot.

3. Get information on system login processes

This is done using the -l option. Consider the following example:

```
$ who -l
LOGIN    tty4          2012-08-07 05:32           1309 id=4
LOGIN    tty5          2012-08-07 05:32           1313 id=5
LOGIN    tty2          2012-08-07 05:32           1322 id=2
LOGIN    tty3          2012-08-07 05:32           1324 id=3
LOGIN    tty6          2012-08-07 05:32           1327 id=6
LOGIN    tty1          2012-08-07 05:32           1492 id=1
```

So we see that information related to system login processes was displayed in the output.

4. Get the hostname and user associated with stdin

This is done using the -m option. Consider the following example:

```
$ who -m
iafzal pts/1      2012-08-07 07:58 (:0.0)
```

So we see that the relevant information was produced in the output.

5. Get the current run level

This is done using the -r option. Consider the following example:

```
$ who -r
run-level 2 2012-08-07 05:32
```

So we see that the information related to current run level (which is 2) was produced in the output.

6. Get the list of user logged in

This is done using the -u option. Consider the following example:

```
$ who -u
iafzal tty7      2012-08-07 05:33 old      1619 (:0)
iafzal pts/0      2012-08-07 06:47 00:31      2336 (:0.0)
iafzal pts/1      2012-08-07 07:58 .      2336 (:0.0)
```

So we see that a list of logged-in users was produced in the output.

7. Get number of users logged-in and their user names

This is done using the -q option. Consider the following example:

```
$ who -q
iafzal iafzal iafzal
# users=3
```

So we see that information related to number of logged-in users and their user names was produced in the output.

8. Get all the information

This is done using the **-a** option. Consider the following example:

```
$ who -a
      system boot 2012-08-07 05:32
      run-level 2 2012-08-07 05:32
LOGIN    tty4      2012-08-07 05:32          1309 id=4
LOGIN    tty5      2012-08-07 05:32          1313 id=5
LOGIN    tty2      2012-08-07 05:32          1322 id=2
LOGIN    tty3      2012-08-07 05:32          1324 id=3
LOGIN    tty6      2012-08-07 05:32          1327 id=6
LOGIN    tty1      2012-08-07 05:32          1492 id=1
iafzal + tty7      2012-08-07 05:33 old      1619 (:0)
iafzal + pts/0      2012-08-07 06:47 .        2336 (:0.0)
iafzal + pts/1      2012-08-07 07:58 .        2336 (:0.0)
```

So we see that all the information that 'who' can print is produced in output.

last command:

To find out when a **particular user last logged in** to the Linux or Unix server.

Syntax

The basic syntax is:

```
last
last [userNameHere]
last [tty]
last [options] [userNameHere]
```

If no options provided last command displays a list of all users logged in (and out). You can filter out results by supplying names of users or terminal to show only those entries matching the username/tty.

last command examples

To find out who has recently logged in and out on your server, type:

```
$ last
```

Sample outputs:

```
root      pts/1      10.1.6.120      Tue Jan 28 05:59      still logged in
root      pts/0      10.1.6.120      Tue Jan 28 04:08      still logged in
root      pts/0      10.1.6.120      Sat Jan 25 06:33 - 08:55  (02:22)
root      pts/1      10.1.6.120      Thu Jan 23 14:47 - 14:51  (00:03)
root      pts/0      10.1.6.120      Thu Jan 23 13:02 - 14:51  (01:48)
```

```
root      pts/0          10.1.6.120      Tue Jan  7 12:02 - 12:38  (00:35)
wtmp begins Tue Jan  7 12:02:54 2014
```

List all users last logged in/out time

last command searches back through the file /var/log/wtmp file and the output may go back to several months. Just use the less command or more command as follows to display output one screen at a time:

```
$ last | more
last | less
```

List a particular user last logged in

To find out when user iafzal last logged in, type:

```
$ last iafzal
$ last iafzal | less
$ last iafzal | grep 'Thu Jan 23'
```

Sample outputs:

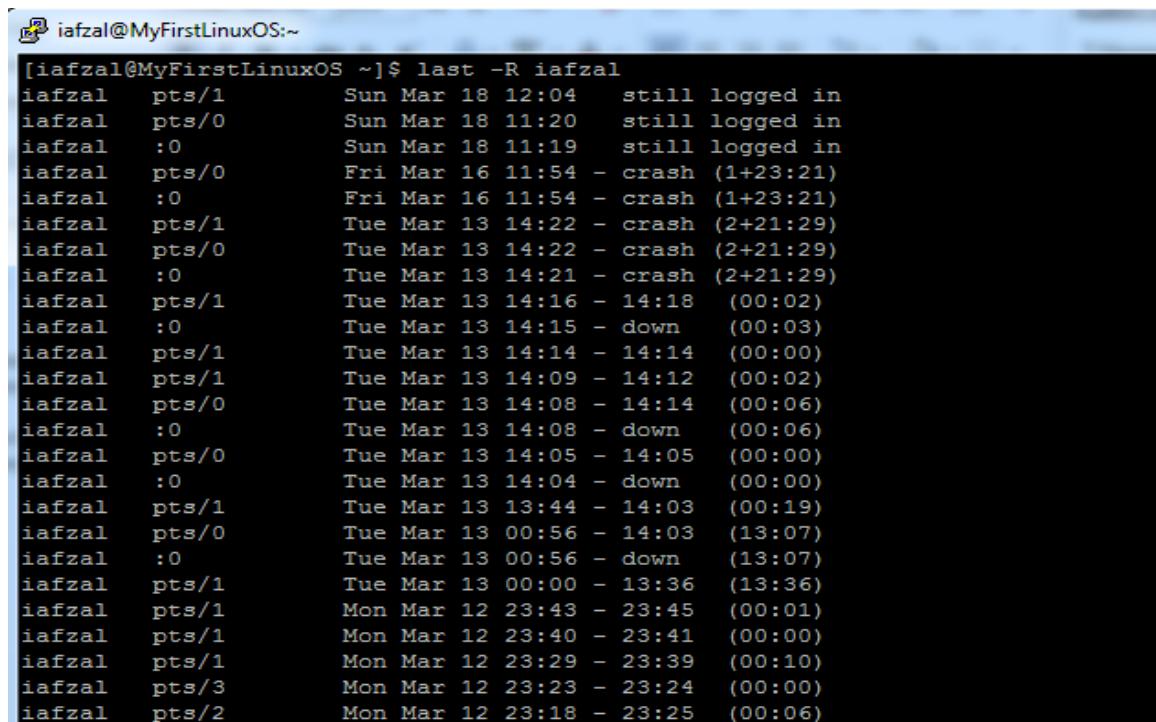
```
[iafzal@MyFirstLinuxOS ~]$ last iafzal
iafzal  pts/1      192.168.56.1      Sun Mar 18 12:04      still logged in
iafzal  pts/0      :0                  Sun Mar 18 11:20      still logged in
iafzal  :0          :0                  Sun Mar 18 11:19      still logged in
iafzal  pts/0      :0                  Fri Mar 16 11:54 - crash (1+23:21)
iafzal  :0          :0                  Fri Mar 16 11:54 - crash (1+23:21)
iafzal  pts/1      :0                  Tue Mar 13 14:22 - crash (2+21:29)
iafzal  pts/0      192.168.56.1      Tue Mar 13 14:22 - crash (2+21:29)
iafzal  :0          :0                  Tue Mar 13 14:21 - crash (2+21:29)
iafzal  pts/1      :0                  Tue Mar 13 14:16 - 14:18 (00:02)
iafzal  :0          :0                  Tue Mar 13 14:15 - down (00:03)
iafzal  pts/1      192.168.56.1      Tue Mar 13 14:14 - 14:14 (00:00)
iafzal  pts/1      192.168.56.1      Tue Mar 13 14:09 - 14:12 (00:02)
iafzal  pts/0      :0                  Tue Mar 13 14:08 - 14:14 (00:06)
iafzal  :0          :0                  Tue Mar 13 14:08 - down (00:06)
iafzal  pts/0      :0                  Tue Mar 13 14:05 - 14:05 (00:00)
iafzal  :0          :0                  Tue Mar 13 14:04 - down (00:00)
iafzal  pts/1      192.168.56.1      Tue Mar 13 13:44 - 14:03 (00:19)
iafzal  pts/0      :0                  Tue Mar 13 00:56 - 14:03 (13:07)
iafzal  :0          :0                  Tue Mar 13 00:56 - down (13:07)
iafzal  pts/1      192.168.56.1      Tue Mar 13 00:00 - 13:36 (13:36)
iafzal  pts/1      :0                  Mon Mar 12 23:43 - 23:45 (00:01)
iafzal  pts/1      :0                  Mon Mar 12 23:40 - 23:41 (00:00)
iafzal  pts/1      192.168.56.1      Mon Mar 12 23:29 - 23:39 (00:10)
iafzal  pts/3      192.168.56.1      Mon Mar 12 23:23 - 23:24 (00:00)
iafzal  pts/2      192.168.56.1      Mon Mar 12 23:18 - 23:25 (00:06)
iafzal  pts/1      192.168.56.1      Mon Mar 12 22:51 - 23:25 (00:34)
iafzal  pts/1      192.168.56.1      Mon Mar 12 18:09 - 21:10 (03:00)
iafzal  pts/0      :0                  Mon Mar 12 18:07 - 13:36 (19:28)
iafzal  :0          :0                  Mon Mar 12 18:07 - crash (06:48)
iafzal  pts/0      :0                  Mon Mar 12 17:55 - 18:06 (00:10)
iafzal  :0          :0                  Mon Mar 12 17:55 - 18:06 (00:10)
iafzal  pts/1      192.168.56.1      Thu Mar  8 14:09 - 14:11 (00:02)
```

Hide hostnames (Linux only)

To hide the display of the hostname field pass -R option:

```
$ last -R  
last -R iafzal
```

Sample outputs:



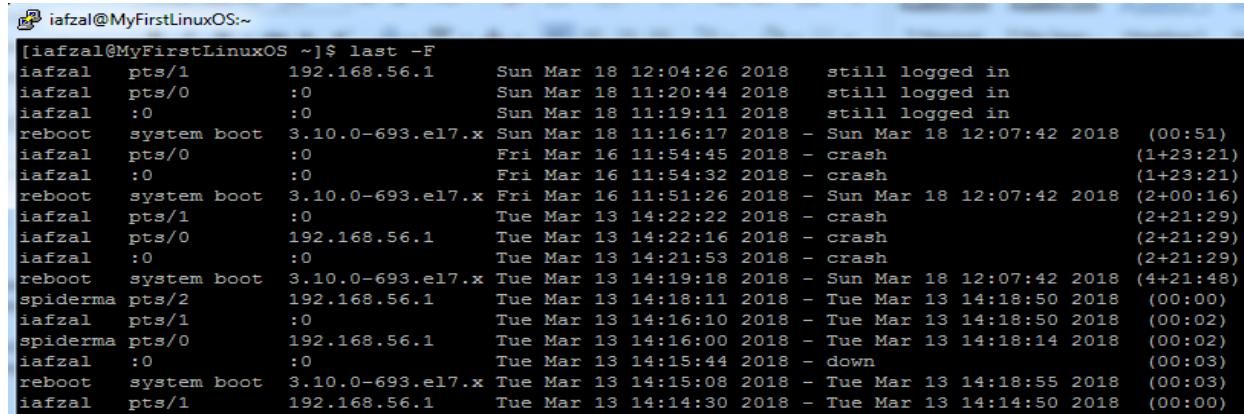
```
[iafzal@MyFirstLinuxOS ~]$ last -R iafzal  
iafzal pts/1 Sun Mar 18 12:04 still logged in  
iafzal pts/0 Sun Mar 18 11:20 still logged in  
iafzal :0 Sun Mar 18 11:19 still logged in  
iafzal pts/0 Fri Mar 16 11:54 - crash (1+23:21)  
iafzal :0 Fri Mar 16 11:54 - crash (1+23:21)  
iafzal pts/1 Tue Mar 13 14:22 - crash (2+21:29)  
iafzal pts/0 Tue Mar 13 14:22 - crash (2+21:29)  
iafzal :0 Tue Mar 13 14:21 - crash (2+21:29)  
iafzal pts/1 Tue Mar 13 14:16 - 14:18 (00:02)  
iafzal :0 Tue Mar 13 14:15 - down (00:03)  
iafzal pts/1 Tue Mar 13 14:14 - 14:14 (00:00)  
iafzal pts/1 Tue Mar 13 14:09 - 14:12 (00:02)  
iafzal pts/0 Tue Mar 13 14:08 - 14:14 (00:06)  
iafzal :0 Tue Mar 13 14:08 - down (00:06)  
iafzal pts/0 Tue Mar 13 14:05 - 14:05 (00:00)  
iafzal :0 Tue Mar 13 14:04 - down (00:00)  
iafzal pts/1 Tue Mar 13 13:44 - 14:03 (00:19)  
iafzal pts/0 Tue Mar 13 00:56 - 14:03 (13:07)  
iafzal :0 Tue Mar 13 00:56 - down (13:07)  
iafzal pts/1 Tue Mar 13 00:00 - 13:36 (13:36)  
iafzal pts/1 Mon Mar 12 23:43 - 23:45 (00:01)  
iafzal pts/1 Mon Mar 12 23:40 - 23:41 (00:00)  
iafzal pts/1 Mon Mar 12 23:29 - 23:39 (00:10)  
iafzal pts/3 Mon Mar 12 23:23 - 23:24 (00:00)  
iafzal pts/2 Mon Mar 12 23:18 - 23:25 (00:06)
```

Display complete login & logout times

By default year is now displayed by last command. You can force last command to display full login and logout times and dates by passing -F option:

```
$ last -F
```

Sample outputs:



```
[iafzal@MyFirstLinuxOS ~]$ last -F  
iafzal pts/1 192.168.56.1 Sun Mar 18 12:04:26 2018 still logged in  
iafzal pts/0 :0 Sun Mar 18 11:20:44 2018 still logged in  
iafzal :0 :0 Sun Mar 18 11:19:11 2018 still logged in  
reboot system boot 3.10.0-693.el7.x Sun Mar 18 11:16:17 2018 - Sun Mar 18 12:07:42 2018 (00:51)  
iafzal pts/0 :0 Fri Mar 16 11:54:45 2018 - crash (1+23:21)  
iafzal :0 :0 Fri Mar 16 11:54:32 2018 - crash (1+23:21)  
reboot system boot 3.10.0-693.el7.x Fri Mar 16 11:51:26 2018 - Sun Mar 18 12:07:42 2018 (2+00:16)  
iafzal pts/1 :0 Tue Mar 13 14:22:22 2018 - crash (2+21:29)  
iafzal pts/0 192.168.56.1 Tue Mar 13 14:22:16 2018 - crash (2+21:29)  
iafzal :0 :0 Tue Mar 13 14:21:53 2018 - crash (2+21:29)  
reboot system boot 3.10.0-693.el7.x Tue Mar 13 14:19:18 2018 - Sun Mar 18 12:07:42 2018 (4+21:48)  
spiderma pts/2 192.168.56.1 Tue Mar 13 14:18:11 2018 - Tue Mar 13 14:18:50 2018 (00:00)  
iafzal pts/1 :0 Tue Mar 13 14:16:10 2018 - Tue Mar 13 14:18:50 2018 (00:02)  
spiderma pts/0 192.168.56.1 Tue Mar 13 14:16:00 2018 - Tue Mar 13 14:18:14 2018 (00:02)  
iafzal :0 :0 Tue Mar 13 14:15:44 2018 - down (00:03)  
reboot system boot 3.10.0-693.el7.x Tue Mar 13 14:15:08 2018 - Tue Mar 13 14:18:55 2018 (00:03)  
iafzal pts/1 192.168.56.1 Tue Mar 13 14:14:30 2018 - Tue Mar 13 14:14:50 2018 (00:00)
```

Display full user/domain names

```
$ last -w
```

Display last reboot time

The user reboot logs in each time the system is rebooted. Thus following command will show a log of all reboots since the log file was created:

```
$ last reboot  
$ last -x reboot
```

Display last shutdown time

Find out the system shutdown entries and run level changes:

```
$ last -x  
$ last -x shutdown
```

Find out who was logged in at a particular time

The syntax is as follows to see the state of logins as of the specified time:

```
$ last -t YYYYMMDDHHMMSS  
$ last -t YYYYMMDDHHMMSS userNameHere
```

w command:

Options:

```
-h, --no-header      do not print header  
-u, --no-current    ignore current process username  
-s, --short         short format  
-f, --from          show remote hostname field  
-o, --old-style     old style output  
-i, --ip-addr       display IP address instead of hostname (if possible)  
  
      --help      display this help and exit  
-V, --version     output version information and exit
```

id command:

Print user and group information for the specified USER,
or (when USER omitted) for the current user.

```
-a           ignore, for compatibility with other versions
-Z, --context print only the security context of the current user
-g, --group   print only the effective group ID
-G, --groups  print all group IDs
-n, --name    print a name instead of a number, for -ugG
-r, --real    print the real ID instead of the effective ID, with -ugG
-u, --user    print only the effective user ID
-z, --zero    delimit entries with NUL characters, not whitespace;
              not permitted in default format
--help       display this help and exit
--version    output version information and exit
```

If-then Scripts:

Check the variable

```
#!/bin/bash

count=100
if [ $count -eq 100 ]
then
    echo Count is 100
else
    echo Count is not 100
fi
```

Check if a file error.txt exist

```
#!/bin/bash

clear
if [ -e /home/iafzal/error.txt ]

    then
        echo "File exist"
    else
        echo "File does not exist"
fi
```

Check if a variable value is met

```
#!/bin/bash

a=`date | awk '{print $1}'`

if [ "$a" == Mon ]

    then
        echo Today is $a
    else
        echo Today is not Monday
fi
```

Check the response and then output

```
#!/bin/bash

clear
echo
echo "What is your name?"
echo
read a
echo

echo Hello $a sir
echo

echo "Do you like working in IT? (y/n)"
read Like
echo

if [ "$Like" == y ]
then
echo You are cool

elif [ "$Like" == n ]
then
echo You should try IT, it's a good field
echo
fi
```

Other If statements

If the output is either Monday or Tuesday

```
if [ "$a" = Monday ] || [ "$a" = Tuesday ]
```

Test if the error.txt file exist and its size is greater than zero

```
if test -s error.txt
```

```
if [ $? -eq 0 ]
```

If input is equal to zero (0)

```
if [ -e /export/home/filename ]
```

If file is there

```
if [ "$a" != "" ]
```

If variable does not match

```
if [ error_code != "0" ]
```

If file not equal to zero (0)

Comparisons:

-eq	equal to for numbers
==	equal to for letters
-ne	not equal to
!==	not equal to for letters
-lt	less than
-le	less than or equal to
-gt	greater than
-ge	greater than or equal to

File Operations:

- s file exists and is not empty
- f file exists and is not a directory
- d directory exists
- x file is executable
- w file is writable
- r file is readable

System Utility Commands:

- **date**
- **uptime**
- **hostname**
- **uname**
- **which**
- **cal**
- **bc**

date

Print or set the system date and time

```
Usage: date [OPTION]... [+FORMAT]
or:  date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]
Display the current time in the given FORMAT, or set the system date.
```

Mandatory arguments to long options are mandatory for short options too.

```
-d, --date=STRING          display time described by STRING, not 'now'
-f, --file=DATEFILE        like --date once for each line of DATEFILE
-I[TIMESPEC], --iso-8601[=TIMESPEC]  output date/time in ISO 8601 format.
                                TIMESPEC='date' for date only (the default),
                                'hours', 'minutes', 'seconds', or 'ns' for date
                                and time to the indicated precision.
-r, --reference=FILE       display the last modification time of FILE
-R, --rfc-2822              output date and time in RFC 2822 format.
Example: Mon, 07 Aug 2006 12:34:56 -0600
--rfc-3339=TIMESPEC        output date and time in RFC 3339 format.
                                TIMESPEC='date', 'seconds', or 'ns' for
                                date and time to the indicated precision.
                                Date and time components are separated by
                                a single space: 2006-08-07 12:34:56-06:00
-s, --set=STRING            set time described by STRING
-u, --utc, --universal      print or set Coordinated Universal Time (UTC)
--help                      display this help and exit
--version                   output version information and exit
```

uptime:

Tell how long the system has been running

uptime gives a one line display of the following information. The current time, how long the system has been running, how many users are currently logged on, and the system load averages for the past 1, 5, and 15 minutes

Options:

```
-p, --pretty    show uptime in pretty format
-h, --help      display this help and exit
```

```
-s, --since      system up since
-V, --version    output version information and exit
```

hostname

Show or set the system's host name

Program options:

-a, --alias	alias names
-A, --all-fqdns	all long host names (FQDNs)
-b, --boot	set default hostname if none available
-d, --domain	DNS domain name
-f, --fqdn, --long	long host name (FQDN)
-F, --file	read host name or NIS domain name from given file
-i, --ip-address	addresses for the host name
-I, --all-ip-addresses	all addresses for the host
-s, --short	short host name
-y, --yp, --nis	NIS/YP domain name

Description:

This command can get or set the host name or the NIS domain name. You can also get the DNS domain or the FQDN (fully qualified domain name).

Unless you are using bind or NIS for host lookups you can change the FQDN (Fully Qualified Domain Name) and the DNS domain name (which is part of the FQDN) in the /etc/hosts file

uname

This command will give you system information. It is one of the important command that should be used every time you login to a Linux/Unix machine.

Usage: uname [OPTION]...

Print certain system information. With no OPTION, same as -s.

-a, --all	print all information, in the following order, except omit -p and -i if unknown:
-s, --kernel-name	print the kernel name
-n, --nodename	print the network node hostname
-r, --kernel-release	print the kernel release
-v, --kernel-version	print the kernel version
-m, --machine	print the machine hardware name
-p, --processor	print the processor type or "unknown"
-i, --hardware-platform	print the hardware platform or "unknown"
-o, --operating-system	print the operating system
--help	display this help and exit
--version	output version information and exit

which

Shows the full path of (shell) commands

Usage: /usr/bin/which [options] [--] COMMAND [...]

Write the full path of COMMAND(s) to standard output.

```
--version, -[vV] Print version and exit successfully.  
--help,           Print this help and exit successfully.  
--skip-dot       Skip directories in PATH that start with a dot.  
--skip-tilde     Skip directories in PATH that start with a tilde.  
--show-dot       Don't expand a dot to current directory in output.  
--show-tilde    Output a tilde for HOME directory for non-root.  
--tty-only      Stop processing options on the right if not on tty.  
--all, -a        Print all matches in PATH, not just the first  
--read-alias, -i Read list of aliases from stdin.  
--skip-alias    Ignore option --read-alias; don't read stdin.  
--read-functions Read shell functions from stdin.  
--skip-functions Ignore option --read-functions; don't read stdin.
```

cal and bc

cal command is simply for calendar and bc is for calculator

Processes

- Whenever you enter a command at the shell prompt, it invokes a program. While this program is running it is called a process. Your login shell is also a process, created for you upon logging in and existing until you logout.
- LINUX is a multi-tasking operating system. Any user can have multiple processes running simultaneously, including multiple login sessions. As you do your work within the login shell, each command creates at least one new process while it executes.
- Process id: every process in a LINUX system has a unique PID - process identifier.
- ps - displays information about processes. Note that the ps command differs between different LINUX systems - see the local ps man page for details.

To see your current shell's processes:

```
% ps
  PID      TTY      TIME CMD
 26450    pts/9    0:00 ps
 66801    pts/9    0:00 -csh
```

To see a detailed list of all of your processes on a machine (current shell and all other shells):

```
% ps uc
USER      PID %CPU %MEM   SZ   RSS      TTY STAT      STIME TIME COMMAND
jsmith    26451  0.0  0.0   120   232  pts/9 R      21:01:14 0:00 ps
jsmith    43520  0.0  1.0   300   660  pts/76 S     19:18:31 0:00 elm
jsmith    66801  0.0  1.0   348   640  pts/9 S     20:49:20 0:00 csh
jsmith   112453  0.0  0.0   340   432  pts/76 S     Mar  03 0:00 csh
```

To see a detailed list of every process on a machine:

```
% ps ug
USER      PID %CPU %MEM   SZ   RSS      TTY STAT      STIME TIME COMMAND
root      0  0.0  0.0    8    8      - S      Feb 08 32:57 swapper
root      1  0.1  0.0   252   188     - S      Feb 08 39:16 /etc/init
root     514 72.6  0.0   12     8      - R      Feb 08 28984:05 kproc
root     771  0.2  0.0   16     16     - S      Feb 08 65:14 kproc
root    1028  0.0  0.0   16     16     - S      Feb 08  0:00 kproc
{ lines deleted }
root   60010  0.0  0.0  1296   536     - S      Mar  07  0:00 -ncd19:0
kdr    60647  0.0  0.0   288   392  pts/87 S     Mar  06  0:00 -ksh
manfield 60968  0.0  0.0   268   200     - S      10:12:52 0:00 mwm
kelly   61334  0.0  0.0   424   640     - S      08:18:10 0:00 twm
sjw    61925  0.0  0.0   552   376     - S      Mar  06  0:00 rlogin kanaha
mkm    62357  0.0  0.0   460   240     - S      Feb 08  0:00 xterm
ishley  62637  0.0  0.0   324   152  pts/106 S     Mar  06  0:00 xedit march2
tusciora 62998  0.0  0.0   340   448     - S      Mar  06  0:05 xterm -e
dilfeath 63564  0.0  0.0   200   268     - S      07:32:45 0:00 xclock
tusciora 63878  0.0  0.0   548   412     - S      Mar  06  0:41 twm
```

- kill - use the kill command to send a signal to a process. In most cases, this will be a kill signal, hence the command name. However, other types of signals are usually supported. Note that you can only kill processes which you own. The command syntax is:

```
kill [-signal] process_identifier(PID)
```

Examples:

```
kill 63878      - kills process 63878
kill -9 1225     - kills (kills!) process 1225. Use if
                  simple kill doesn't work.
kill -STOP 2339   - stops process 2339
kill -CONT 2339   - continues stopped process 2339
kill -l           - list the supported kill signals
```

You can also use CTRL-C to kill the currently running process.

- Suspend a process: Use CTRL-Z.
- Background a process: Normally, commands operate in the foreground - you can not do additional work until the command completes. Backgrounding a command allows you to continue working at the shell prompt.

To start a job in the background, use an ampersand (&) when you invoke the command:

```
myprog &
```

To put an already running job in the background, first suspend it with CRTL-Z and then use the "bg" command:

```
myprog      - execute a process
CTRL-Z       - suspend the process
bg          - put suspended process in background
```

- Foreground a process: To move a background job to the foreground, find its "job" number and then use the "fg" command. In this example, the jobs command shows that two processes are running in the background. The fg command is used to bring the second job (%2) to the foreground.

```
jobs
[1] + Running      xcalc
[2]   Running      find / -name core -print
fg %2
```

- Stop a job running in the background: Use the jobs command to find its job number, and then use the stop command. You can then bring it to the foreground or restart execution later.

```
jobs
[1] + Running      xcalc
[2]   Running      find / -name core -print
stop %2
```

- Kill a job running in the background, use the jobs command to find its job number, and then use the kill command. Note that you can also use the ps and kill commands to accomplish the same task.

```
jobs
[1] + Running      xcalc
[2]   Running      find / -name core -print
kill %2
```

- Some notes about background processes:
 - If a background job tries to read from the terminal, it will automatically be stopped by the shell. If this happens, you must put it in the foreground to supply the input.
 - The shell will warn you if you attempt to logout and jobs are still running in the background. You can then use the jobs command to review the list of jobs and act accordingly. Alternately, you can simply issue the logout command again and you will be permitted to exit

Linux Programs

A program, or command, interacts with the kernel to provide the environment and perform the functions called for by the user. A program can be: an executable shell file, known as a shell script; a built-in shell command; or a source compiled, object code file.

The shell is a command line interpreter. The user interacts with the kernel through the shell. You can write ASCII (text) scripts to be acted upon by a shell.

System programs are usually binary, having been compiled from C source code. These are located in places like /bin, /usr/bin, /usr/local/bin, /usr/ucb, etc. They provide the functions that you normally think of when you think of Linux. Some of these are sh, csh, date, who, more, and there are many others.

crontab – Quick Reference

crontab is used to schedule task/jobs

Setting up cron jobs in Unix, Solaris & Linux

cron is a Unix, solaris, Linux utility that allows tasks to be automatically run in the background at regular intervals by the cron daemon.

cron meaning – There is no definitive explanation but most accepted answers is reportedly from Ken Thompson (author of unix cron), name cron comes from chron ,the Greek prefix for ‘time.’.

What is cron ? – Cron is a daemon which runs at the times of system boot from /etc/init.d scripts. If needed it can be stopped/started/restart using init script or with command service crond start in Linux systems.

This document covers following aspects of Unix, Linux cron jobs to help you understand and implement cronjobs successfully

1. What is crontab?
2. What is a cron job or cron schedule?
3. Crontab Restrictions
4. Crontab Commands
5. Crontab file – syntax
6. Crontab Example
7. Crontab Environment
8. Disable Email
9. Generate log file for crontab activity
10. Crontab file location

1. What is crontab?

Crontab (CRON TABLE) is a file which contains the schedule of cron entries to be run and at specified times. File location varies by operating systems, See Crontab file location at the end of this document.

2.What is a cron job or cron schedule?

Cron job or cron schedule is a specific set of execution instructions specifying day, time and command to execute. crontab can have multiple execution statements.

3. Crontab Restrictions

You can execute crontab if your name appears in the file /usr/lib/cron/cron.allow. If that file does not exist, you can use crontab if your name does not appear in the file /usr/lib/cron/cron.deny. If only cron.deny exists and is empty, all users can use crontab. If neither file exists, only the root user can use crontab. The allow/deny files consist of one user name per line.

4. Crontab Commands

export EDITOR=vi ;to specify a editor to open crontab file.

crontab -e Edit crontab file, or create one if it doesn't already exist.
crontab -l crontab list of cronjobs , display crontab file contents.
crontab -r Remove your crontab file.
crontab -v Display the last time you edited your crontab file. (This option is only available on a few systems.)

5. Crontab file

Crontab syntax :

A crontab file has five fields for specifying day , date and time followed by the command to be run at that interval.

*	*	*	*	*	command to be executed
-	-	-	-	-	
				-----	day of week (0 - 6)
(Sunday=0)					
			-----	month (1 - 12)	
		+-----	day of	month (1 - 31)	
	+-----	hour (0 - 23)			
+-----	min (0 - 59)				

* in the value field above means all legal values as in braces for that column.

The value column can have a * or a list of elements separated by commas. An element is either a number in the ranges shown above or two numbers in the range separated by a hyphen (meaning an inclusive range).

Notes

A.) Repeat pattern like /2 for every 2 minutes or /10 for every 10 minutes is not supported by all operating systems. If you try to use it and crontab complains it is probably not supported.

B.) The specification of days can be made in two fields: month day and weekday. If both are specified in an entry, they are cumulative meaning both of the entries will get executed .

6. Crontab Examples

A line in crontab file like below removes the tmp files from /home/someuser/tmp each day at 6:30 PM.

```
30 18 * * * rm /home/someuser/tmp/*
```

Changing the parameter values as below will cause this command to run at different time schedule below :

min	hour	day/month	month	day/week	Execution time
30	0	1	1,6,12	*	— 00:30 Hrs on 1st of Jan, June & Dec.
0	20	*	10	1-5	— 8.00 PM every weekday (Mon-Fri) only in Oct.
0	0	1,10,15	*	*	— midnight on 1st ,10th & 15th of month
5,10	0	10	*	1	— At 12.05,12.10 every Monday & on 10th of every month

Note : If you inadvertently enter the crontab command with no argument(s), do not attempt to get out with Control-d. This removes all entries in your crontab file. Instead, exit with Control-c.

7. Crontab Environment

cron invokes the command from the user's HOME directory with the shell, (/usr/bin/sh). cron supplies a default environment for every shell, defining:

HOME=user's-home-directory

LOGNAME=user's-login-id

PATH=/usr/bin:/usr/sbin:..

SHELL=/usr/bin/sh

Users who desire to have their .profile executed must explicitly do so in the crontab entry or in a script called by the entry.

8. Disable Email

By default cron jobs sends a email to the user account executing the cronjob. If this is not needed put the following command At the end of the cron job line .

```
>/dev/null 2>&1
```

9. Generate log file

To collect the cron execution execution log in a file :

```
30 18 * * * rm /home/someuser/tmp/* > /home/someuser/cronlogs/clean_tmp_dir.log
```

10. Crontab file location

User crontab files are stored by the login names in different locations in different Unix and Linux flavors. These files are useful for backing up, viewing and restoring but should be edited only with crontab command by the users.

- **Mac OS X**
/usr/lib/cron/tabs/
- **BSD Unix**
/var/cron/tabs/
- **Solaris, HP-UX, Debian, Ubuntu**
/var/spool/cron/crontabs/
- **AIX, Red Hat Linux, CentOS, Ferdora**
/var/spool/cron/

System Resources Commands:

Command/Syntax	What it will do
date	report the current date and time
df	report the summary of disk blocks and inodes free and in use
du	report amount of disk space in use+
hostname/uname	display or set (super-user only) the name of the current machine
passwd	set or change your password
whereis	report the binary, source, and man page locations for the command
which	reports the path to the command or the shell alias in use
who or w	report who is logged in and what processes are running
cal	displays a calendar
bc	Calculator

df - summarize disk block and file usage

df is used to report the number of disk blocks and inodes used and free for each file system. The output format and valid options are very specific to the OS and program version in use.

Syntax

df [options] [resource]

Common Options

- l local file systems only (SVR4)
- k report in kilobytes (SVR4)

du - report disk space in use

du reports the amount of disk space in use for the files or directories you specify.

Syntax

du [options] [directory or file]

Common Options

- a display disk usage for each file, not just subdirectories
- s display a summary total only
- k report in kilobytes (SVR4)

who - list current users

who reports who is logged in at the present time.

Syntax

who [am i]

Examples

```
> who
wmtell ttyp1 Apr 21 20:15 (apple.acs.ohio-s)
fbwalk ttyp2 Apr 21 23:21 (worf.acs.ohio-st)
stwang ttyp3 Apr 21 23:22 (127.99.25.8)
```

whereis - report program locations

whereis reports the filenames of source, binary, and manual page files associated with command(s).

Syntax

whereis [options] command(s)

Common Options

- b report binary files only
- m report manual sections only
- s report source files only

Examples

```
> whereis Mail
Mail: /usr/ucb/Mail /usr/lib/Mail.help /usr/lib/Mail.rc /usr/man/man1/Mail.1
> whereis -b Mail
Mail: /usr/ucb/Mail /usr/lib/Mail.help /usr/lib/Mail.rc
> whereis -m Mail
Mail: /usr/man/man1/Mail.1
```

which - report the command found

which will report the name of the file that is be executed when the command is invoked. This will be the full path name or the alias that's found first in your path.

Syntax

which command(s)

example--

```
> which Mail
/usr/ucb/Mail
```

hostname/uname -n = name of machine

hostname (uname -n on SysV) reports the host name of the machine the user is logged into, e.g.:

```
> hostname
yourcomputername
```

uname has additional options to print information about system hardware type and software version.

date - current date and time

date displays the current date and time. A superuser can set the date and time.

Syntax

date [options] [+format]

Common Options

-u use Universal Time (or Greenwich Mean Time)

+format specify the output format

%a weekday abbreviation, Sun to Sat

%h month abbreviation, Jan to Dec

%j day of year, 001 to 366

%n <new-line>

%t <TAB>

%y last 2 digits of year, 00 to 99

%D MM/DD/YY date

%H hour, 00 to 23

%M minute, 00 to 59

%S second, 00 to 59

%T HH:MM:SS time

Examples

> date

Mon Jun 10 09:01:05 EDT 1996

> date -u

Mon Jun 10 13:01:33 GMT 1996

> date +%a%t%D

Mon 06/10/96

> date '+%y:%j'

96:162

Terminal Control Keys

- Several key combinations on your keyboard usually have a special effect on the terminal.
- These "control" (CTRL) keys are accomplished by holding the CTRL key while typing the second key. For example, CTRL-c means to hold the CTRL key while you type the letter "c".
- The most common control keys are listed below:

CTRL-u - erase everything you've typed on the command line

CTRL-c - stop/kill a command

CTRL-h - backspace (usually)

CTRL-z - suspend a command

CTRL-s - stop the screen from scrolling

CTRL-q - continue scrolling

CTRL-d - exit from an interactive program (signals end of data)

top command

Know what is happening in “real time” on your systems is in my opinion the basis to use and optimize your OS. The `top` command can help us, this is a very useful system monitor that is really easy to use, and that can also allows us to understand why our OS suffers and which process use most resources. The command to be run on the terminal is:

```
$ top
```

And we’ll get a screen similar to the one on the right:

Let’s see now every single row of this output to explain all the information found within the screen.

1° Row — top

```
top - 11:37:19 up 1 day, 1:25, 3 users, load average: 0.02, 0.12, 0.07
```

This first line indicates in order:

- current time (11:37:19)
- uptime of the machine (up 1 day, 1:25)
- users sessions logged in (3 users)
- average load on the system (load average: 0.02, 0.12, 0.07) the 3 values refer to the last minute, five minutes and 15 minutes.

2° Row – task

```
Tasks: 73 total, 2 running, 71 sleeping, 0 stopped, 0 zombie
```

The second row gives the following information:

- Processes running in totals (73 total)
- Processes running (2 running)
- Processes sleeping (71 sleeping)
- Processes stopped (0 stopped)
- Processes waiting to be stoppati from the parent process (0 zombie)

3° Row – cpu

```
Cpu(s): 0.3%us, 0.0%sy, 0.0%ni, 99.4%id, 0.0%wa, 0.3%hi, 0.0%si, 0.0%st
```

The third line indicates how the cpu is used. If you sum up all the percentages the total will be 100% of the cpu. Let's see what these values indicate in order:

- Percentage of the CPU for user processes (0.3%**us**)
- Percentage of the CPU for system processes (0.0%**sy**)
- Percentage of the CPU processes with priority upgrade *nice* (0.0%**ni**)
- Percentage of the CPU not used (99,4%**id**)
- Percentage of the CPU processes waiting for I/O operations(0.0%**wa**)
- Percentage of the CPU serving hardware interrupts (0.3% **hi** — Hardware IRQ)
- Percentage of the CPU serving software interrupts (0.0% **si** — Software Interrupts)
- The amount of CPU 'stolen' from this virtual machine by the hypervisor for other tasks (such as running another virtual machine) this will be 0 on desktop and server without Virtual machine. (0.0%**st** — Steal Time)

4° and 5° Rows – memory usage

```
Mem: 2057720k total, 778560k used, 1279160k free, 31976k buffers
Swap: 4192956k total, 68k used, 4192888k free, 563772k cached
```

The fourth and fifth rows respectively indicate the use of physical memory (RAM) and swap. In this order: Total memory in use, free, buffers cached.

Following Rows — Processes list

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
4522	root	15	0	132m	14m	3204	S	0.3	0.7	0:17.75	bb_monitor.pl
1	root	15	0	10328	692	580	S	0.0	0.0	0:01.28	init

And as last thing ordered by CPU usage (as default) there are the processes currently in use. Let's see what information we can get in the different columns:

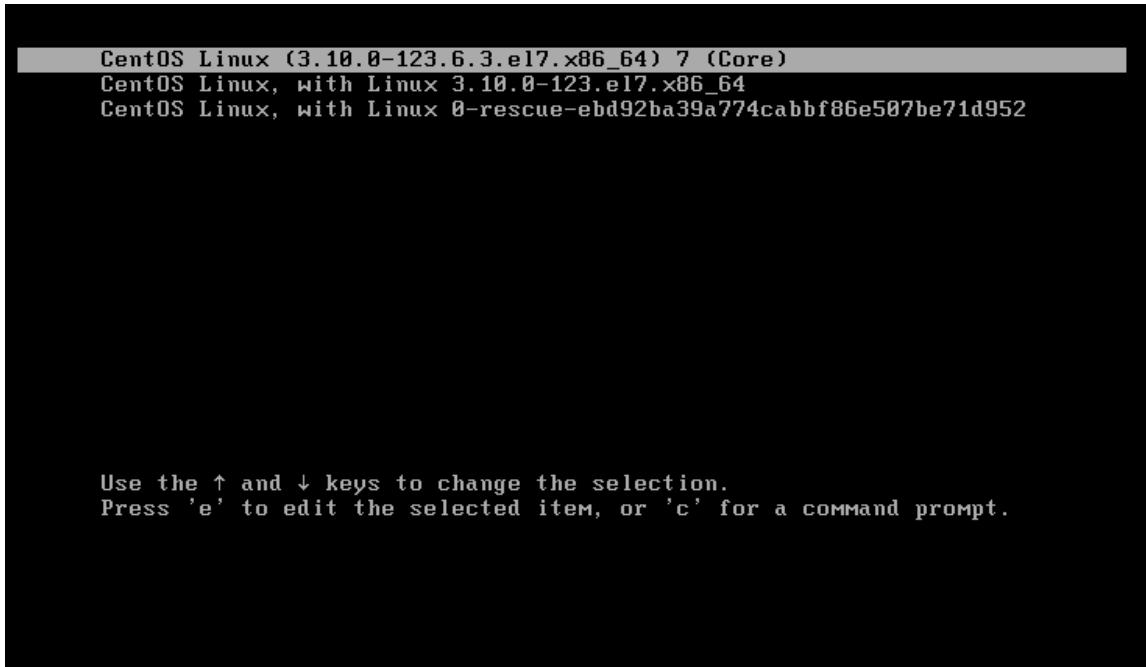
- **PID** – l'ID of the process(4522)
- **USER** – The user that is the owner of the process (root)
- **PR** – priority of the process (15)
- **NI** – The “NICE” value of the process (0)
- **VIRT** – virtual memory used by the process (132m)
- **RES** – physical memory used from the process (14m)
- **SHR** – shared memory of the process (3204)
- **S** – indicates the status of the process: **S**=sleep **R**=running **Z**=zombie (S)
- **%CPU** – This is the percentage of CPU used by this process (0.3)
- **%MEM** – This is the percentage of RAM used by the process (0.7)
- **TIME+** –This is the total time of activity of this process (0:17.75)
- **COMMAND** – And this is the name of the process (bb_monitor.pl)

Conclusions

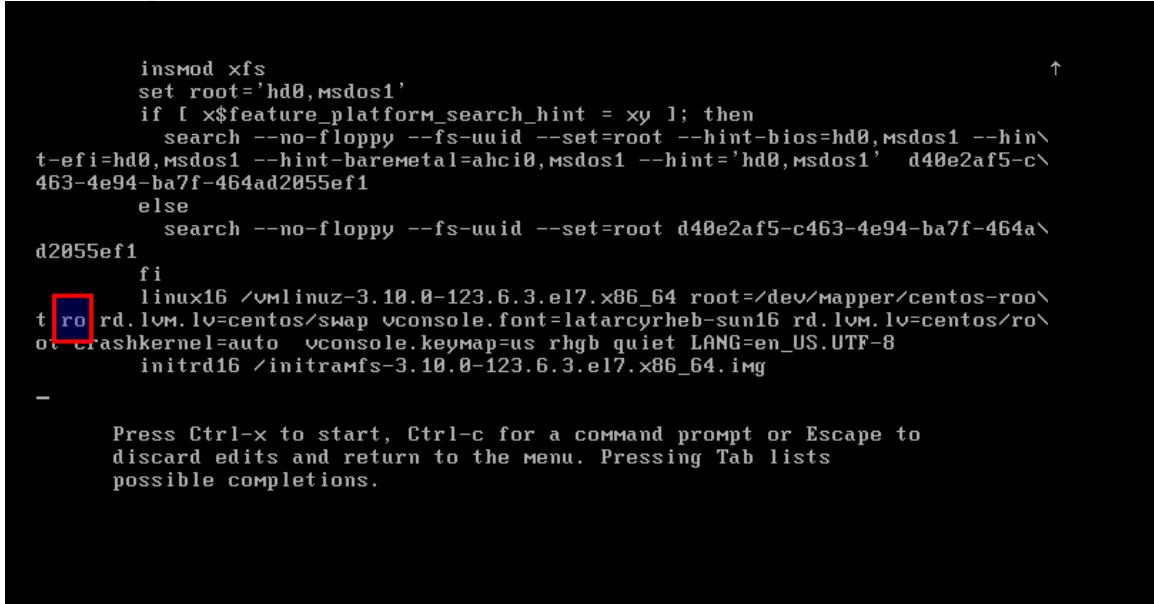
Now that we have seen in detail all the information that the command “top” returns, it will be easier to understand the reason of excessive load and/or the slowing of the system

Recover/Reset Root Password

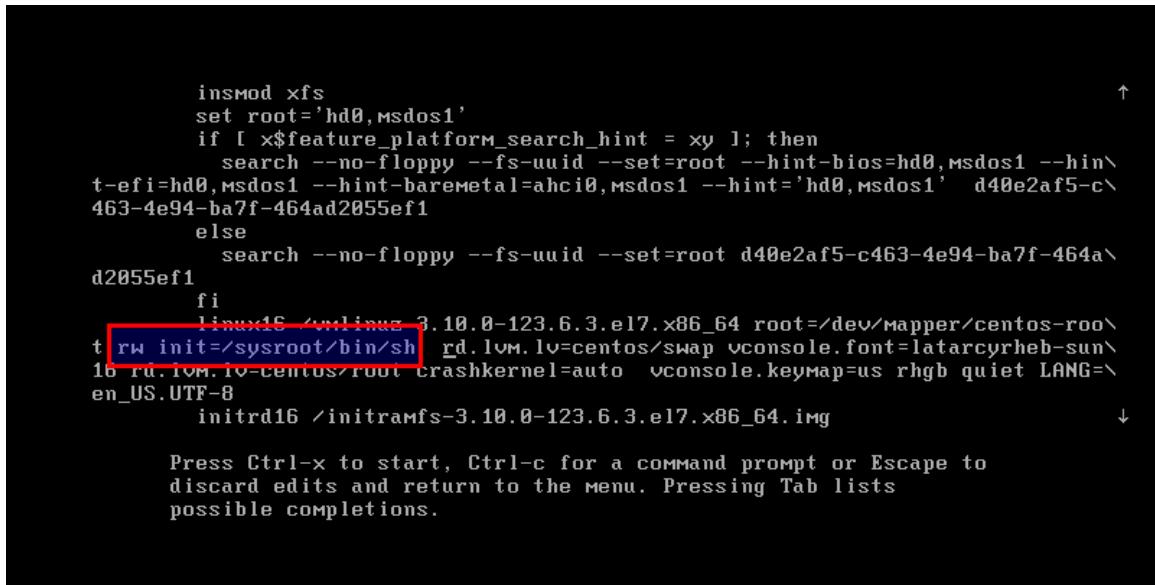
1 – In the boot grub menu select option to edit



2 – Select Option to edit (e).



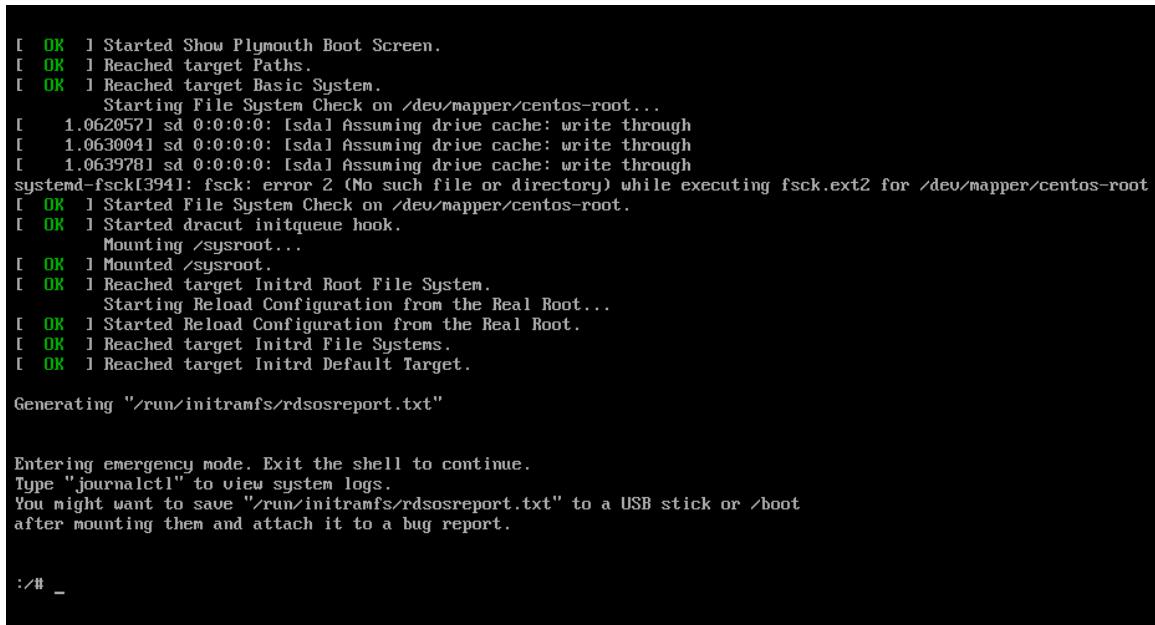
3 – Go to the line = ro and change it with rw init=/sysroot/bin/sh



```
insmod xfs
set root='hd0,msdos1'
if [ $feature_platform_search_hint = xy ]; then
    search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint\x
t-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1' d40e2af5-c\
463-4e94-ba7f-464ad2055ef1
else
    search --no-floppy --fs-uuid --set=root d40e2af5-c463-4e94-ba7f-464a\
d2055ef1
fi
linux16 /vmlinuz-3.10.0-123.6.3.el7.x86_64 root=/dev/mapper/centos-roo\
t rw init=/sysroot/bin/sh rd.lvm.lv=centos/swap vconsole.font=latacyrheb-sun\
16_ru.lvm.lv-centos/root crashkernel=auto vconsole.keymap=us rhgb quiet LANG=\
en_US.UTF-8
initrd16 /initramfs-3.10.0-123.6.3.el7.x86_64.img

Press Ctrl-x to start, Ctrl-c for a command prompt or Escape to
discard edits and return to the menu. Pressing Tab lists
possible completions.
```

4 – Now press Control+x to start on single user mode



```
[ OK ] Started Show Plymouth Boot Screen.
[ OK ] Reached target Paths.
[ OK ] Reached target Basic System.
      Starting File System Check on /dev/mapper/centos-root...
[ 1.062057] sd 0:0:0:0: [sda] Assuming drive cache: write through
[ 1.063004] sd 0:0:0:0: [sda] Assuming drive cache: write through
[ 1.063978] sd 0:0:0:0: [sda] Assuming drive cache: write through
systemd-fsck[3941]: fsck: error 2 (No such file or directory) while executing fsck.ext2 for /dev/mapper/centos-root
[ OK ] Started File System Check on /dev/mapper/centos-root.
[ OK ] Started dracut initqueue hook.
      Mounting /sysroot...
[ OK ] Mounted /sysroot.
[ OK ] Reached target Initrd Root File System.
      Starting Reload Configuration from the Real Root...
[ OK ] Started Reload Configuration from the Real Root.
[ OK ] Reached target Initrd File Systems.
[ OK ] Reached target Initrd Default Target.

Generating "/run/initramfs/rdsosreport.txt"

Entering emergency mode. Exit the shell to continue.
Type "journalctl" to view system logs.
You might want to save "/run/initramfs/rdsosreport.txt" to a USB stick or /boot
after mounting them and attach it to a bug report.

#: _
```

5 – Now access the system with this command.

chroot /sysroot

6 – Reset the password.

passwd root

7 – Exit chroot

exit

8 - Reboot your system

reboot

HANDOUTS

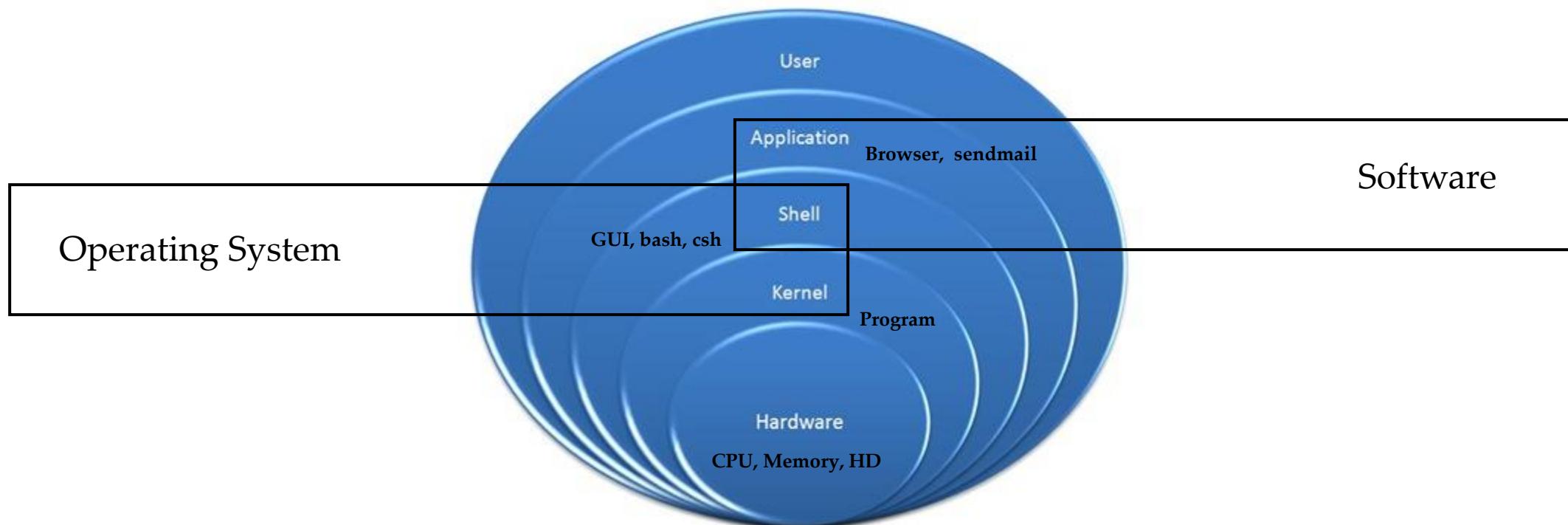
MODULE 6

WELCOME To: MODULE 6

SHELL SCRIPTING

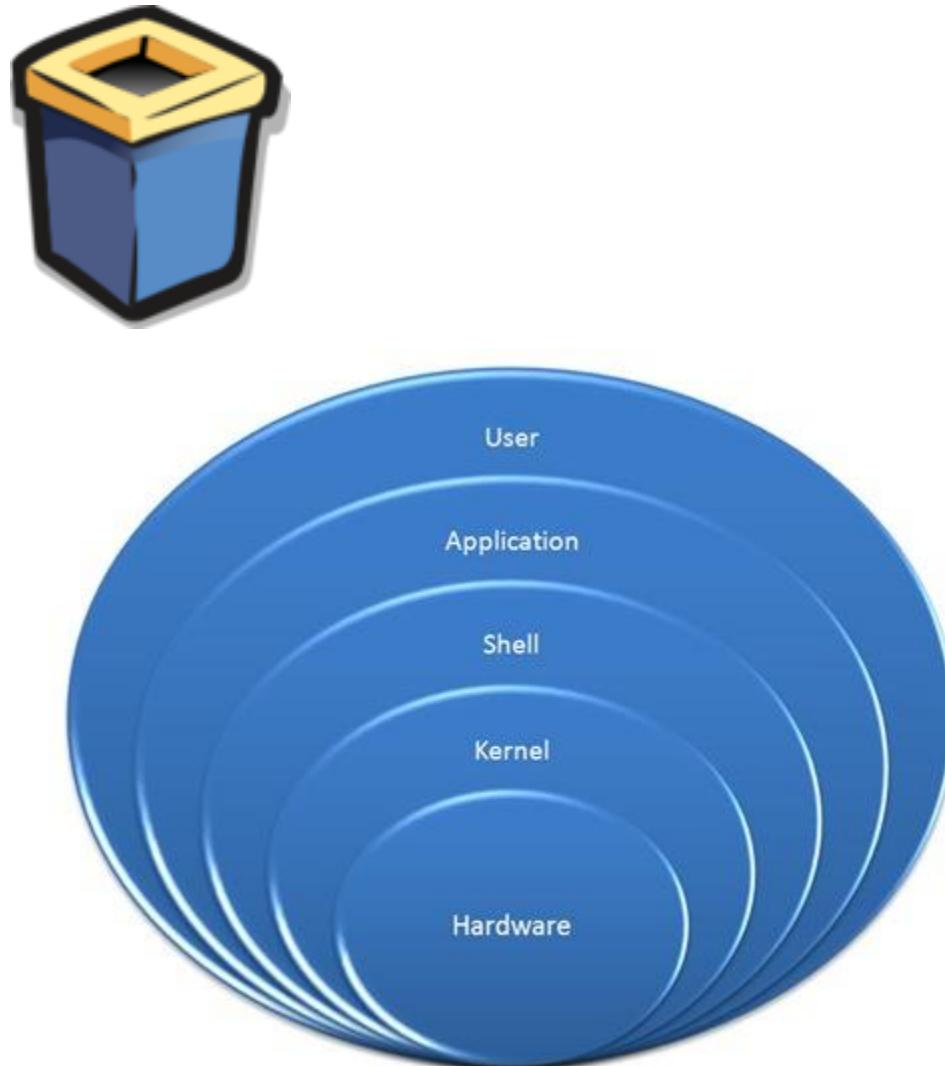
Linux Kernel

- What is a Kernel?
 - Interface between hardware and Software



Introduction to Shell

- What is a Shell?
 - Its like a container
 - Interface between users and Kernel/OS
 - CLI is a Shell
- Find your Shell
 - echo \$0
 - Available Shells “cat /etc/shells”
 - Your Shell? /etc/passwd
- Windows GUI is a shell
- Linux KDE GUI is a shell
- Linux sh, bash etc. is a shell



Types of Shell

- sh
- bash
- ksh
- csh

Starting a Shell

- Type shell name e.g. csh
- Type exit to exit out of shell

Shell Scripting

- What is a Shell Script?

A shell script is an executable file containing multiple shell commands that are executed sequentially. The file can contain:

- Shell (**`#!/bin/bash`**)
- Comments (**`# comments`**)
- Commands (**`echo, cp, grep`** etc.)
- Statements (**`if, while, for`** etc.)
- Shell script should have executable permissions (e.g. **`-rwx r-x r-x`**)
- Shell script has to be called from absolute path (e.g **`/home/userdir/script.bash`**)
- If called from current location then **`./script.bash`**

Shell Script – Basic Scripts

- Output to screen using “echo”
- Creating tasks
 - Telling your id, current location, your files/directories, system info
 - Creating files or directories
 - Output to a file “>”
- Filters/Text processors through scripts (**cut**, **awk**, **grep**, **sort**, **uniq**, **wc**)

Input and Output of Script

- Create script to take input from the user

read

echo

if-then Scripts

- If then statement

If this happens = do this

Otherwise = do that

For Loop Scripts

- For loops

Keep running until specified number of variable

e.g: `variable = 10` then run the script 10 times

OR

`variable = green, blue, red` (then run the script 3 times for each color.)

do-while Scripts

- do while

The while statement continually executes a block of statements while a particular condition is true or met

e.g: Run a script until 2pm

```
while [ condition ]  
do  
    command1  
    command2  
    commandN  
done
```

Case Statement Scripts

- Case

If option a is selected = do this

If option b is selected = do this

If option c is selected = do this.

Check Other Servers Connectivity

- A script to check the status of remote hosts

Aliases

- Aliases is a very popular command that is used to cut down on lengthy and repetitive commands

```
alias ls="ls -al"  
alias pl="pwd; ls"  
alias tell="whoami; hostname; pwd"  
alias dir="ls -l | grep ^d"  
alias lmar="ls -l | grep Mar"  
alias wpa= "chmod a+w"  
alias d="df -h | awk '{print \$6}' | cut -c1-4"
```

Creating User or Global Aliases

- User = Applies only to a specific user profile
 - Global = Applies to everyone who has account on the system
-
- User = **/home/user/.bashrc**
 - Global = **/etc/bashrc**

```
alias hh="hostname"
```

Shell History

- Command “history”

UNIX Kernel:

Technically speaking, the UNIX kernel "is" the operating system. It provides the basic full time software connection to the hardware. By full time, it means that the kernel is always running while the computer is turned on. When a system boots up, kernel is loaded. Likewise, the kernel is only exited when the computer is turned off.

The UNIX kernel is built specifically for a machine when it is installed. It has a record of all the pieces of hardware it needs to talk to and knows what languages they speak (how to turn switches on and off to get a desired result). Thus, a kernel is not easily ported to another computer. Each individual computer will have its own tailor-made kernel. If the computer's hardware configuration changes during its life, the kernel must be "rebuilt" (told about the new pieces of hardware).

However, though the connection between the kernel and the hardware is "hardcoded" to a specific machine, the connection between the user and the kernel is generic. That is the beauty of the UNIX kernel. From your perspective, regardless of how the kernel interacts with the hardware, no matter which UNIX computer you use, you will have the same kernel interface to work with. That is because the hardware is "hidden" by the kernel.

The kernel also handles memory management, input and output requests, and process scheduling for time-shared operations (we'll talk more about what this means later).

To help it with its work, the kernel also executes daemon programs which stay alive as long as the machine is turned on and help perform tasks such as printing or serving web documents.

However, the task of hiding the hardware is a pretty much full time job for the kernel. As such, it does not have too much time to provide for a fancy user-friendly interface. Thus, though the kernel is much easier to talk to than the hardware, the language of the kernel is still pretty cryptic.

Fortunately, the UNIX operating system has built in "shells" which wrap around the kernel and provide a much more user-friendly interface. Let's take a look at shells.

Shells

The shell sits between you and the kernel, acting as a command interpreter. It reads your terminal input and translates the commands into actions taken by the system. The shell is analogous to command in DOS. When you log into the system you are given a default shell. When the shell starts up it reads its startup files and may set environment variables, command search paths, and command aliases, and executes any commands specified in these files.

The original shell was the Bourne shell, sh. Every Linux platform will either have the Bourne shell, or a Bourne compatible shell available. It has very good features for controlling input and output, but is not well suited for the interactive user. To meet the latter need the C shell, csh, was written and is now found on most, but not all, Linux systems. It uses C type syntax, the language Unix is written in, but has a more awkward input/output implementation. It has job control, so that you can reattach a job running in the background to the foreground. It also provides a history feature which allows you to modify and repeat previously executed commands.

The default prompt for the Bourne shell is \$ (or #, for the root user). The default prompt for C shell is %.

Numerous other shells are available from the network. Almost all of them are based on either sh or csh with extensions to provide job control to sh, allow in-line editing of commands, page through previously executed commands, provide command name completion and custom prompt, etc. Some of the more well known of these may be on your favorite Linux system: the Korn shell, ksh, by David Korn and the Bourne Again Shell, bash, from the Free Software Foundations GNU project, both based on sh, the T-C shell, tcsh, and the extended C shell, cshe, both based on csh. Below we will describe some of the features of sh and csh so that you can get started.

Built-in Commands

The shells have a number of built-in, or native commands. These commands are executed directly in the shell and don't have to call another program to be run. These built-in commands are different for the different shells.

sh

For the Bourne shell some of the more commonly used built-in commands are:

- : null command
- . source (read and execute) commands from a file
- case case conditional loop
- cd change the working directory (default is \$HOME)
- echo write a string to standard output
- eval evaluate the given arguments and feed the result back to the shell
- exec execute the given command, replacing the current shell

exit exit the current shell
export share the specified environment variable with subsequent shells
for for conditional loop
if if conditional loop
pwd print the current working directory
read read a line of input from stdin
set set variables for the shell
test evaluate an expression as true or false
trap trap for a typed signal and execute commands
umask set a default file permission mask for new files
unset unset shell variables
wait wait for a specified process to terminate
while while conditional loop

csh

For the C shell the more commonly used built-in functions are:

alias assign a name to a function
bg put a job into the background
cd change the current working directory
echo write a string to stdout
eval evaluate the given arguments and feed the result back to the shell
exec execute the given command, replacing the current shell
exit exit the current shell
fg bring a job to the foreground
foreach for conditional loop
glob do filename expansion on the list, but no "\\" escapes are honored
history print the command history of the shell
if if conditional loop
jobs list or control active jobs
kill kill the specified process
limit set limits on system resources
logout terminate the login shell
nice command lower the scheduling priority of the process, command
nohup command do not terminate command when the shell exits
set set a shell variable
setenv set an environment variable for this and subsequent shells
stop stop the specified background job
umask set a default file permission mask for new files
unalias remove the specified alias name
unset unset shell variables
while while conditional loop

Environment Variables

Environmental variables are used to provide information to the programs you use. You can have both global environment and local shell variables. Global environment variables are set by your login shell and new programs and shells inherit the environment of their parent shell. Local shell variables are used only by that shell and are not passed on to other processes. A child process cannot pass a variable back to its parent process.

The current environment variables are displayed with the "env" or "printenv" commands. Some common ones are:

- DISPLAY The graphical display to use, e.g. nyssa:0.0
- EDITOR The path to your default editor, e.g. /usr/bin/vi
- GROUP Your login group, e.g. staff
- HOME Path to your home directory, e.g. /home/frank
- HOST The hostname of your system, e.g. nyssa
- IFS Internal field separators, usually any white space (defaults to tab, space and <newline>)
- LOGNAME The name you login with, e.g. frank
- PATH Paths to be searched for commands, e.g. /usr/bin:/usr/ucb:/usr/local/bin
- PS1 The primary prompt string, Bourne shell only (defaults to \$)
- PS2 The secondary prompt string, Bourne shell only (defaults to >)
- SHELL The login shell you're using, e.g. /usr/bin/csh
- TERM Your terminal type, e.g. xterm
- USER Your username, e.g. frank

Many environment variables will be set automatically when you login. You can modify them or define others with entries in your startup files or at any time within the shell. Some variables you might want to change are PATH and DISPLAY. The PATH variable specifies the directories to be automatically searched for the command you specify. Examples of this are in the shell startup scripts below.

You set a global environment variable with a command similar to the following for the C shell:

```
% setenv NAME value
```

and for Bourne shell:

```
$ NAME=value; export NAME
```

You can list your global environmental variables with the env or printenv commands. You unset them with the unsetenv (C shell) or unset (Bourne shell) commands.

To set a local shell variable use the set command with the syntax below for C shell. Without options set displays all the local variables.

```
% set name=value
```

For the Bourne shell set the variable with the syntax:

```
$ name=value
```

The current value of the variable is accessed via the "\$name", or "\${name}", notation

Shell

Whenever you login to a Linux system you are placed in a shell program. The shell's prompt is usually visible at the cursor's position on your screen. To get your work done, you enter commands at this prompt.

The shell is a command interpreter; it takes each command and passes it to the operating system kernel to be acted upon. It then displays the results of this operation on your screen.

Several shells are usually available on any UNIX system, each with its own strengths and weaknesses.

Different users may use different shells. Initially, your system administrator will supply a default shell, which can be overridden or changed. The most commonly available shells are:

Bourne shell (sh)

- C shell (csh)
- Korn shell (ksh)
- TC Shell (tcsh)
- Bourne Again Shell (bash)

Each shell also includes its own programming language. Command files, called "shell scripts" are used to accomplish a series of tasks.

Shell Script Execution

Once a shell script is created, there are several ways to execute it. However, before any Korn shell script can be executed, it must be assigned the proper permissions. The chmod command changes permissions on individual files, in this case giving execute permission to the simple file:

```
$ chmod +x simple
```

After that, you can execute the script by specifying the filename as an argument to the bash command:

```
$ bash simple
```

You can also execute scripts by just typing its name alone. However, for that method to work, the directory containing the script must be defined in your PATH variable. When looking at your .profile earlier in the course, you may have noticed that the PATH=\$PATH:\$HOME definition was already in place. This enables you to run scripts located in your home directory (\$HOME) without using the ksh command. For instance, because of that pre-defined PATH variable, the simple script can be run from the command line like this:

```
$ simple
```

(For the purposes of this course, we'll simplify things by running all scripts by their script name only, not as an argument to the ksh command.)

You can also invoke the script from your current shell by opening a background subprocess - or subshell - where the actual command processing will occur. You won't see it running, but it will free up your existing shell so you can continue working. This is really only necessary when running long, processing-intensive scripts that would otherwise take over your current shell until they complete.

To run the script you created in the background, invoke it this way:

```
$ simple &
```

When the script completes, you'll see output similar to this in the current shell:

```
[1] - Done (127) simple
```

It is important to understand that Korn shell scripts run in a somewhat different way than they would in other shells. Specifically, variables defined in the Korn shell aren't understood outside of the defining - or parent - shell. They must be explicitly exported from the parent shell to work in a subsequent script or subshell. If you use the export or typeset -x commands to make the variable known outside the parent shell, any subshell will automatically inherit the values you defined for the parent shell.

For example, here's a script named lookmeup that does nothing more than print a line to standard output using the myaddress (defined as 123 Anystreet USA) variable:

```
$ cat lookmeup  
print "I live at $myaddress"
```

If you open a new shell (using the ksh command) from the parent shell and run the script, you see that myaddress is undefined:

```
$ ksh  
$ lookmeup  
I live at
```

```
$
```

However, if you export myaddress from the parent shell:

```
$ exit  
$ export myaddress
```

and then open a new shell and run the lookmeup script again, the variable is now defined:

```
$ ksh  
$ lookmeup  
I live at 123 Anystreet USA
```

To illustrate further how the parent shell takes processing precedence, let's change the value of myaddress in the subshell:

```
$ myaddress='Houston, Texas'  
  
$ print $myaddress  
Houston, Texas
```

Now, if you exit the new shell and go back to the parent shell and type the same command:

```
$ exit  
$ print $myaddress  
123 Anystreet USA
```

you see that the original value in the parent shell was not affected by what you did in the subshell.

A way to export variables automatically is to use the set -o allexport command. This command cannot export variables to the parent shell from a subshell, but can export variables created in the parent shell to all subshells created after the command is run. Likewise, it can automatically export variables created in subshells to new subshells created after running the command. set -o allexport is a handy command to place in your .kshrc file.

Shell Scripts - An Illustrated View

At the risk of sounding redundant, let's recap: shell scripts are simply files containing a list of commands to be executed in sequence. Now let's go a bit further and look at a shell script, line-by-line.

Any Korn shell script should contain this line at the very beginning:

```
#!/usr/bin/ksh
```

As you probably already know, the # sign marks anything that follows it on the line as a comment - anything coming after it won't be interpreted or processed as part of the script. But, when the # character is followed by a ! (commonly called "bang"), the meaning changes. The line above specifies that the Korn shell will be (or should be) executing the script. If nothing is specified, the system will attempt to execute the script using whatever its default shell type is, not necessarily a Korn shell. Since the Korn shell supports some commands that other shells do not, this can sometimes cause a problem. To be valid, this line must be on the very first line of the script.

Shell scripts are often used to automate day-to-day tasks. For example, a system administrator might use the following script, named diskuse here, to keep track of disk space usage:

```
#!/usr/bin/ksh
# diskuse
# Shows disk usage in blocks for /home
cd /var/log
cp disk.log disk.log.0
cd /home
```

```
du -sk * > /var/log/disk.log  
cat /var/log/disk.log
```

Shown again - but this time with annotation - the script's processing steps are clear:

```
#!/usr/bin/ksh  
# SCRIPT NAME: diskuse  
# SCRIPT PURPOSE: Shows disk usage in blocks for /home  
  
# change to the directory where disk.log resides  
cd /var/log  
  
# make a copy of disk.log  
cp disk.log disk.log.0  
  
# change to the target directory  
cd /home  
  
# run the du -sk * command on all files  
# in /home and redirect the output  
# to /var/log/disk.log  
du -sk * > /var/log/disk.log  
  
# display the output of the du -sk *  
# command to standard output  
cat /var/log/disk.log
```

It's not a good idea to hard-code pathnames into your scripts like we did in the previous example. We specified /var/log as the target directory several times, but what if the location of the files changed? In a short script like this one, the impact is not great. However, some scripts can be hundreds of lines long, creating a maintenance headache if files are moved. A way around this is to create a variable to take the place of the full pathname, such as:

```
LOGDIR=/var/log
```

The fourth line of the script would change from:

```
cp disk.log disk.log.0
```

to:

```
cp ${LOGDIR}/disk.log ${LOGDIR}/disk.log.0
```

Then, if the locations of disk.log changes in the future, you would only have to change the variable definition to update the script. Also note that since you are defining the pathname with the LOGDIR variable, the cd /var/log line in the script is unnecessary. Likewise, the du -sk * > /var/log/disk.log and cat /var/log/disk.log lines can substitute \${LOGDIR} for /var/log.

Basic Shell Scripts:

Output to screen

```
#!/bin/bash
# Simple output script

echo "Hello World"
```

Defining Tasks

```
#!/bin/bash
# Define small tasks

whoami
echo
pwd
echo
hostname
echo
ls -ltr
echo
```

Defining variables

```
#!/bin/bash
# Example of defining variables

a=Imran
b=Afzal
c='Linux class'

echo "My first name is $a"
echo "My surname is $b"
echo 'My surname is $c'
```

Read Input

```
#!/bin/bash
# Read user input

echo "What is your first name?"
read a
echo

echo "What is your last name?"
```

```
read b
echo

echo Hello $a $b
```

Scripts to run commands within

```
#!/bin/bash
# Script to run commands within

clear
echo "Hello `whoami`"
echo
echo "Today is `date`"
echo
echo "Number of user login: `who | wc -l`"
echo
```

Read input and perform a task

```
#!/bin/bash
# This script will rename a file

echo Enter the file name to be renamed
read oldfilename

echo Enter the new file name
read newfilename

mv $oldfilename $newfilename
echo The file has been renamed as $newfilename
```

for loop Scripts:

Simple for loop output

```
#!/bin/bash

for i in 1 2 3 4 5
do
echo "Welcome $i times"
done
```

Simple for loop output

```
#!/bin/bash

for i in eat run jump play
do
echo See Imran $i
done
```

for loop to create 5 files named 1-5

```
#!/bin/bash

for i in {1..5}
do
touch $i
done
```

for loop to delete 5 files named 1-5

```
#!/bin/bash

for i in {1..5}
do
rm $i
done
```

Specify days in for loop

```
#!/bin/bash
```

```
i=1
```

```
for day in Mon Tue Wed Thu Fri
do
    echo "Weekday $((i++)) : $day"
done
```

List all users one by one from /etc/passwd file

```
#!/bin/bash

i=1
for username in `awk -F: '{print $1}' /etc/passwd`
do
    echo "Username $((i++)) : $username"
done
```

do-while Script

Script to run for a number of times

```
#!/bin/bash

c=1
while [ $c -le 5 ]
do
    echo "Welcone $c times"
    (( c++ ))
done
```

Script to run for a number of seconds

```
#!/bin/bash

count=0
num=10
while [ $count -lt 10 ]
do
    echo
    echo $num seconds left to stop this process $1
    echo
    sleep 1

num=`expr $num - 1`
count=`expr $count + 1`
done
echo
echo $1 process is stopped!!!
echo
```

If-then Scripts:

Check the variable

```
#!/bin/bash

count=100
if [ $count -eq 100 ]
then
    echo Count is 100
else
    echo Count is not 100
fi
```

Check if a file error.txt exist

```
#!/bin/bash

clear
if [ -e /home/iafzal/error.txt ]

    then
        echo "File exist"
    else
        echo "File does not exist"
fi
```

Check if a variable value is met

```
#!/bin/bash

a=`date | awk '{print $1}'`

if [ "$a" == Mon ]

    then
        echo Today is $a
    else
        echo Today is not Monday
fi
```

Check the response and then output

```
#!/bin/bash

clear
echo
echo "What is your name?"
echo
read a
echo

echo Hello $a sir
echo

echo "Do you like working in IT? (y/n)"
read Like
echo

if [ "$Like" == y ]
then
echo You are cool

elif [ "$Like" == n ]
then
echo You should try IT, it's a good field
echo
fi
```

Other If statements

If the output is either Monday or Tuesday

```
if [ "$a" = Monday ] || [ "$a" = Tuesday ]
```

Test if the error.txt file exist and its size is greater than zero

```
if test -s error.txt
```

```
if [ $? -eq 0 ]
```

If input is equal to zero (0)

```
if [ -e /export/home/filename ]
```

If file is there

```
if [ "$a" != "" ]
```

If variable does not match

```
if [ error_code != "0" ]
```

If file not equal to zero (0)

Comparisons:

-eq	equal to for numbers
==	equal to for letters
-ne	not equal to
!==	not equal to for letters
-lt	less than
-le	less than or equal to
-gt	greater than
-ge	greater than or equal to

File Operations:

- s file exists and is not empty
- f file exists and is not a directory
- d directory exists
- x file is executable
- w file is writable
- r file is readable

case Scripts:

```
#!/bin/bash

echo
echo Please chose one of the options below
echo
echo 'a = Display Date and Time'
echo 'b = List file and directories'
echo 'c = List users logged in'
echo 'd = Check System uptime'
echo

read choices

case $choices in

a) date;;
b) ls;;
c) who;;
d) uptime;;
*) echo Invalid choice - Bye.

esac
```

This script will look at your current day and tell you the state of the backup

```
#!/bin/bash

NOW=$(date +"%a")
case $NOW in
    Mon)
        echo "Full backup";;
    Tue|Wed|Thu|Fri)
        echo "Partial backup";;
    Sat|Sun)
        echo "No backup";;
    *) ;;
esac
```

Aliases

- The alias command allows you to define new commands. Useful for creating shortcuts for longer commands. The syntax is.

```
alias alias-name=executed_command
```

Some examples:

```
alias m=more
alias rm="rm -i"
alias h="history -r | more"
```

To view all current aliases:

```
alias
```

To remove a previously defined alias:

```
unalias alias_name
```

HANDOUTS

MODULE 7

WELCOME TO: MODULE 7

**NETWORKING, SERVICES
AND SYSTEM UPDATES**

Internet Access to VM

- Open **Virtualbox Manager**
- Select the machine you cannot get internet on in the left pane
- Click the **Settings** button in the top menu
- Click **Network** in the left pane in the settings window
- Switched to **Bridged Adaptor** in the **Attached to** drop-down menu
- Hit **OK** to save your changes
- Start your VM

Network Components

- IP
 - Subnet mask
 - Gateway
 - Static vs. DHCP
-
- Interface
 - Interface MAC.

Network Files and Commands

- Interface Detection
- Assigning an IP address
- Interface configuration files
 - /etc/nsswitch.conf
 - /etc/hostname
 - /etc/sysconfig/network
 - /etc/sysconfig/network-scripts/ifcfg-nic
 - /etc/resolv.conf
- Network Commands
 - **ping**
 - **ifconfig**
 - **ifup or ifdown**
 - **netstat**
 - **tcpdump**

NIC Information

NIC = Network Interface Card

Example:

```
ethtool enp0s3
```



Other NICs

lo = The loopback device is a special interface that your computer uses to communicate with itself. It is used mainly for diagnostics and troubleshooting, and to connect to servers running on the local machine

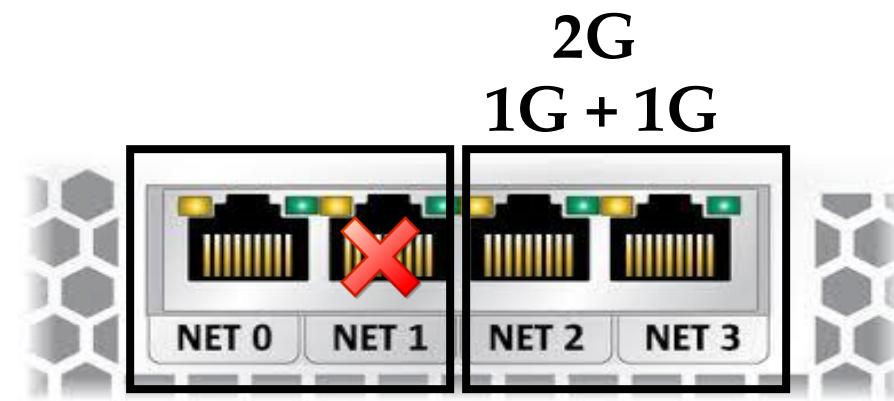
virbr0 = The virbr0, or "Virtual Bridge 0" interface is used for NAT (Network Address Translation). Virtual environments sometimes use it to connect to the outside network

NIC Bonding

NIC = Network Interface Card (PC or laptop)



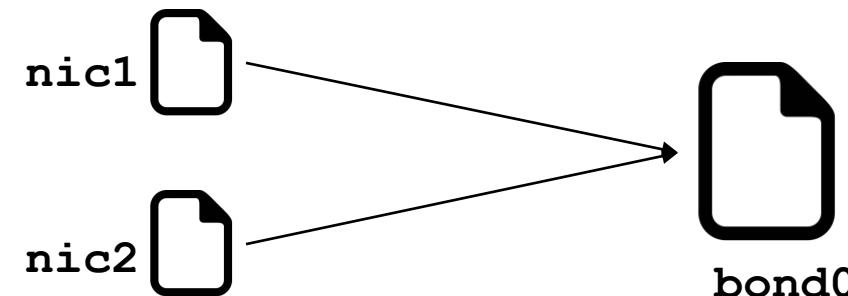
NIC(Network Interface Card) bonding is also known as **Network bonding**. It can be defined as the aggregation or combination of multiple NIC into a single bond interface.



It's main purpose is to provide high availability and redundancy

NIC Bonding Procedure

- modprobe bonding
- modinfo bonding
- Create /etc/sysconfig/network-scripts/ifcfg-bond0
- Edit /etc/sysconfig/network-scripts/ethernet1
- Edit /etc/sysconfig/network-scripts/ethernet2



- Restart network = systemctl restart network

New Network Utilities

What we will learn in this lecture...

- Getting started with **NetworkManager**
- Network configuration methods
 - **nmtui**
 - **nmcli**
 - **nm-connection-editor**
 - **GNOME Settings**.

New Network Utilities

✓ Getting started with **NetworkManager**

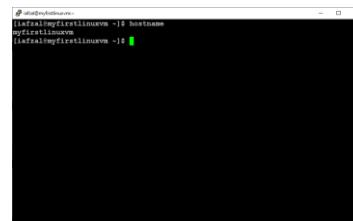
- NetworkManager is a service that provides set of tools designed specifically to make it easier to manage the networking configuration on Linux systems and is the default network management service on RHEL 8
- It makes network management easier
- It provides easy setup of connection to the user
- NetworkManager offers management through different tools such as **GUI**, **nmtui**, and **nmcli**.

New Network Utilities

✓ Network configuration methods

- **nmcli** – Short for network manager command line interface. This tool is useful when access to a graphical environment is not available and can also be used within scripts to make network configuration changes
- **nmtui** – Short for network manager text user interface. This tool can be run within any terminal window and allows changes to be made by making menu selections and entering data
- **nm-connection-editor** - A full graphical management tool providing access to most of the NetworkManager configuration options. It can only be accessed through the desktop or console
- **GNOME Settings** - The network screen of the GNOME desktop settings application allows basic network management tasks to be performed

- Let's practice in our Linux machine...



System Updates and Repos

- yum (CentOS), apt-get (other Linux)
- rpm (Redhat Package Manager)

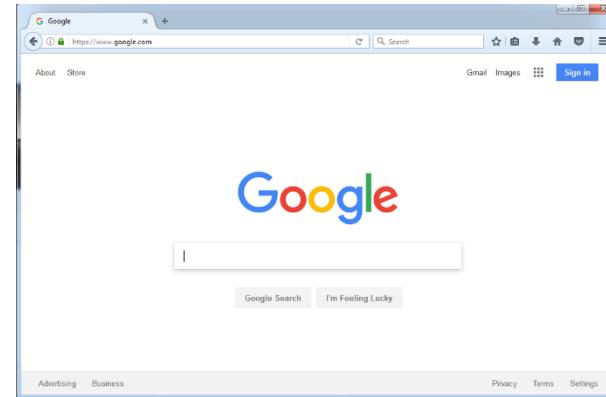
Advance Package Management

- Installing packages
- Upgrading
- Deleting
- View package details information
- Identify source or location information
- Packages configuration files



Download Files or Apps

- Example of Windows browser



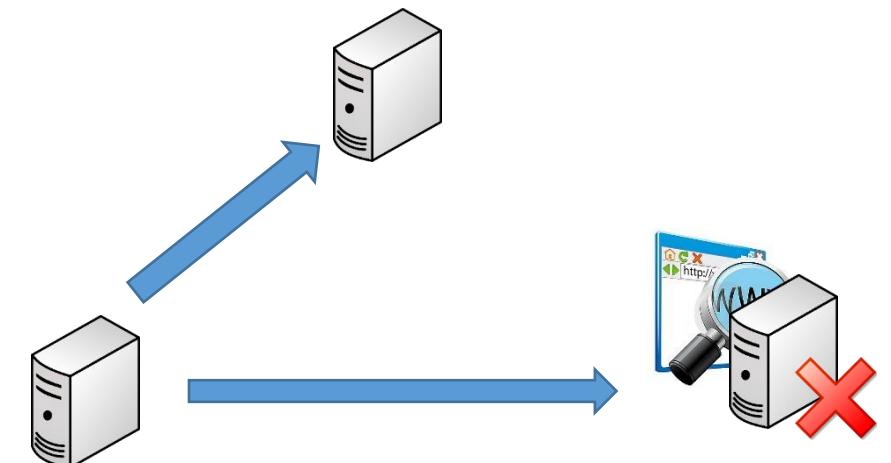
- Linux = **wget**

- Example in Linux:

wget http://website.com/filename

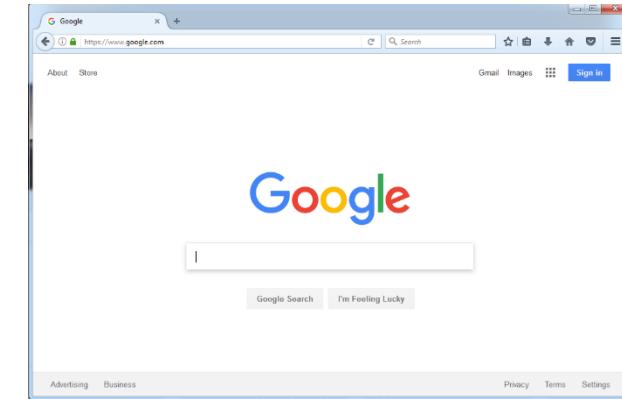
- Why???

Most of the servers in corporate environment do **NOT** have internet access



curl and ping Commands

- Example of Windows browser



- Linux = curl
- Linux = ping

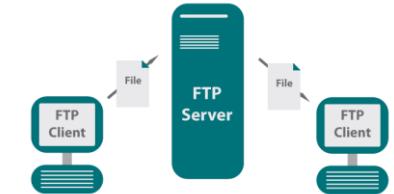
- Example in Linux:

```
curl http://website.com/filename
curl -O http://website.com/filename
```

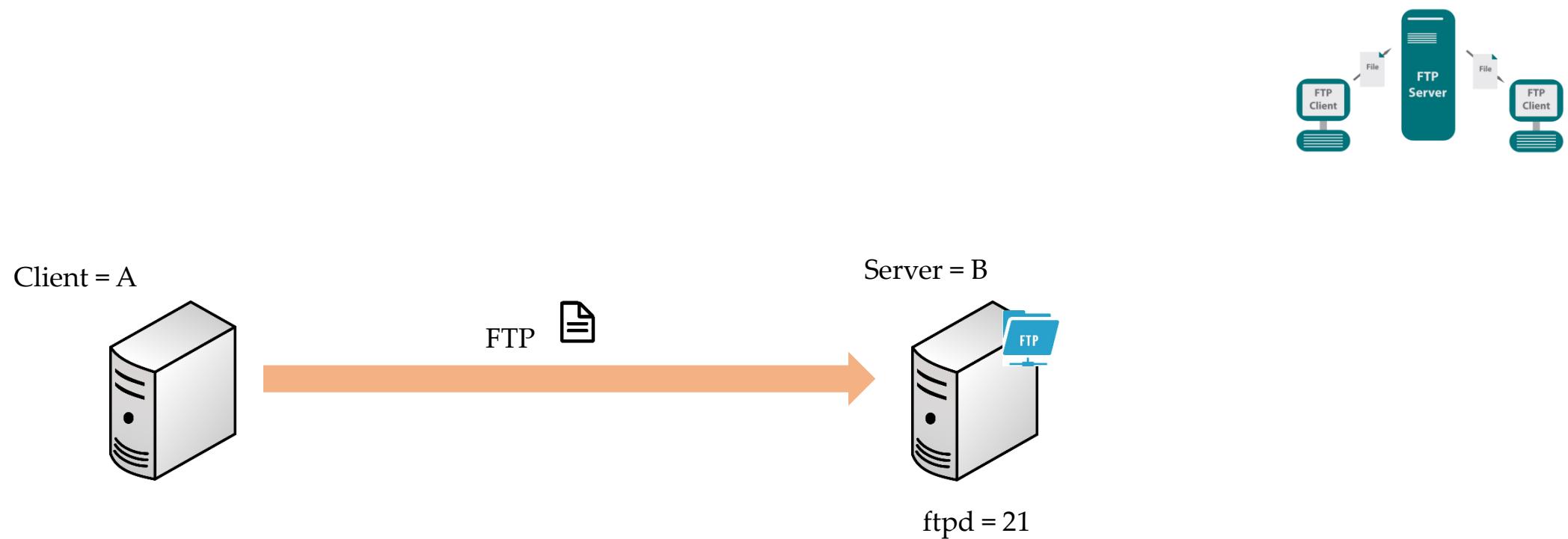
```
ping www.google.com
```

FTP – File Transfer Protocol

- The File Transfer Protocol is a standard network protocol used for the transfer of computer files between a client and server on a computer network. FTP is built on a client-server model architecture using separate control and data connections between the client and the server. (*Wikipedia*)
- Protocol = Set of rules used by computers to communicate
- Default FTP Port = 21
- For this lecture we need 2 Linux machines
 - **Client = MyFirstLinuxVM**
 - **Server = LinuxCentOS7**



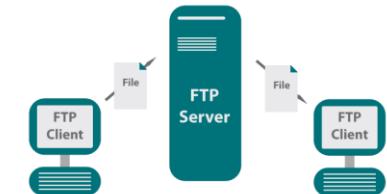
FTP – File Transfer Protocol



FTP – File Transfer Protocol

- **Install and Configure FTP on the remote server**

- # Become root
 - # rpm -qa | grep ftp
 - # ping www.google.com
 - # yum install vsftpd
 - # vi /etc/vsftpd/vsftpd.conf *(make a copy first)*
 - Find the following lines and make the changes as shown below:
 - ## Disable anonymous login ##
 - *anonymous_enable=NO*
 - ## Uncomment ##
 - *ascii_upload_enable=YES*
 - *ascii_download_enable=YES*
 - ## Uncomment - Enter your Welcome message - This is optional ##
 - *ftpd_banner=Welcome to UNIXMEN FTP service.*
 - ## Add at the end of this file ##
 - *use_localtime=YES*
 - # systemctl start vsftpd
 - # systemctl enable vsftpd
 - # systemctl stop firewalld
 - # systemctl disable firewalld
 - # useradd iafzal *(if the user does not exist).*



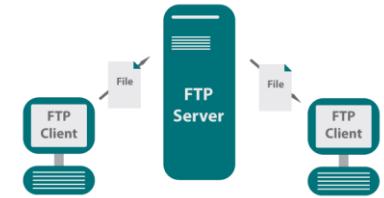
FTP – File Transfer Protocol

- **Install FTP client on the client server**

- # Become root
- # yum install ftp
- # su - iafzal
- \$ touch kruger

- **Commands to transfer file to the FTP server:**

- ftp 192.168.1.x
- Enter username and password
- bi
- hash
- put kruger
- bye.

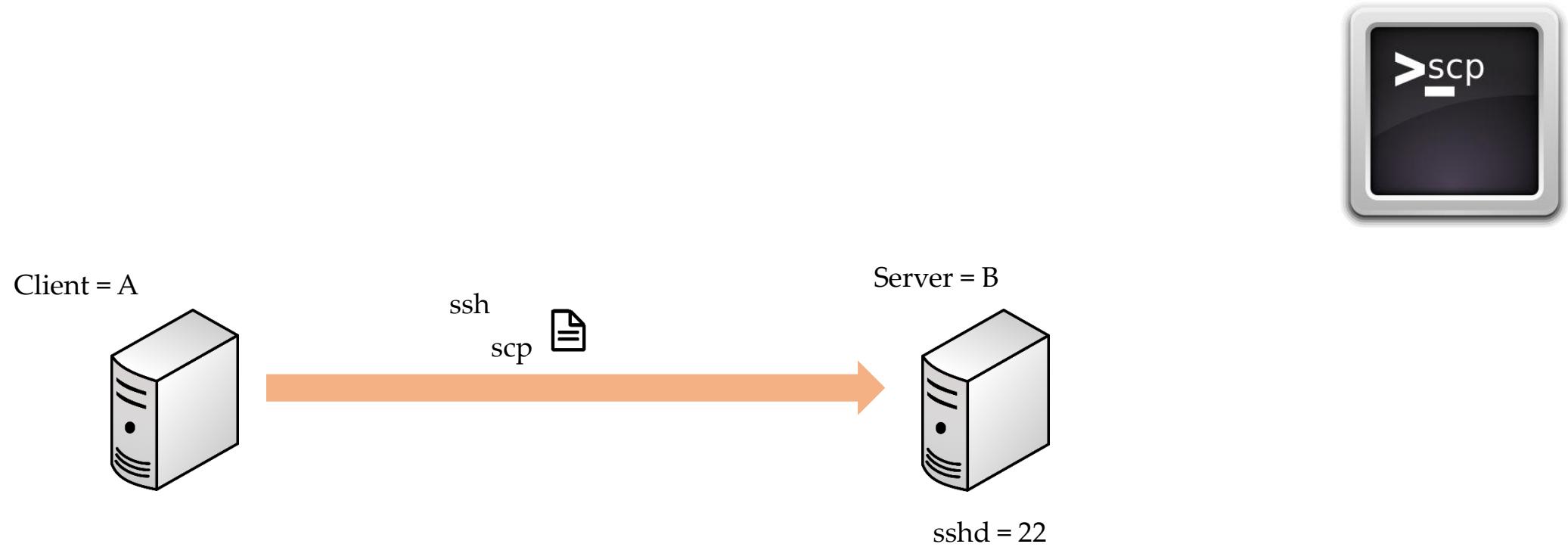


SCP – Secure Copy Protocol

- The Secure Copy Protocol or “SCP” helps to transfer computer files securely from a local to a remote host. It is somewhat similar to the File Transfer Protocol “FTP”, but it adds security and authentication
- Protocol = Set of rules used by computers to communicate
- Default SCP Port = 22 (same as SSH)
- For this lecture we need 2 Linux machines
 - Client = MyFirstLinuxVM
 - Server = LinuxCentOS7



SCP – Secure Copy



SCP – Secure Copy

- SCP commands to transfer file to the remote server:

- Login as yourself (iafzal)
- touch jack
- scp jack iafzal@192.168.1.x:/home/iafzal
- Enter username and password

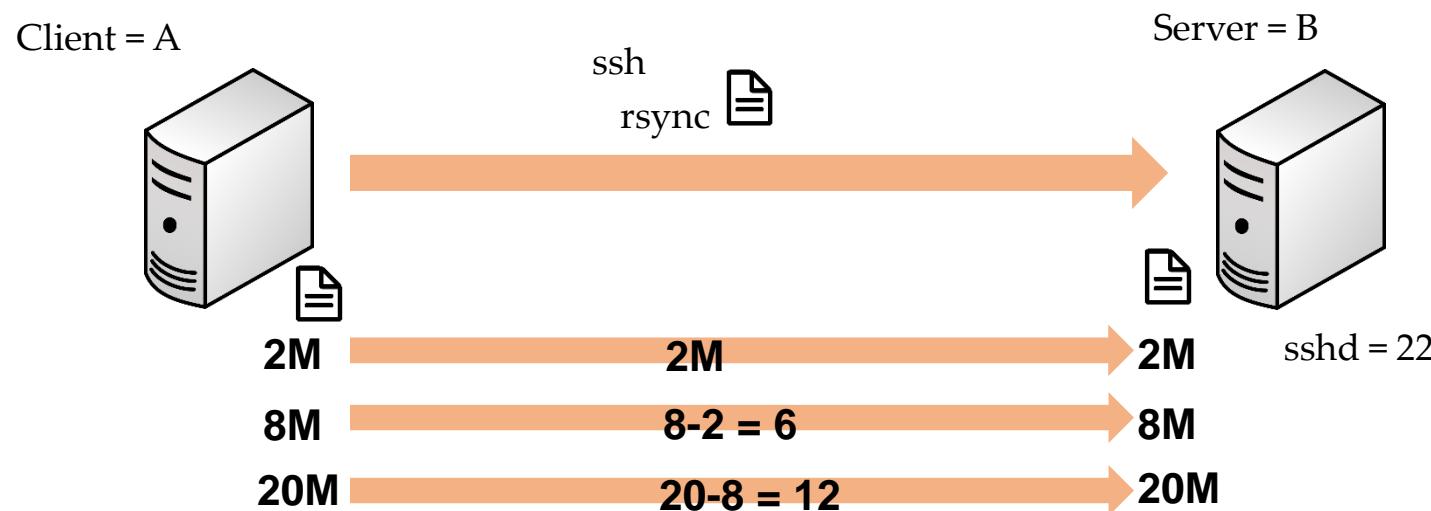


rsync – Remote Synchronization

- **rsync** is a utility for efficiently transferring and synchronizing files within the same computer or to a remote computer by comparing the modification times and sizes of files
- rsync is a lot faster than ftp or scp
- This utility is mostly used to backup the files and directories from one server to another
- Default rsync Port = 22 (same as SSH)
- For this lecture we need 2 Linux machines
 - Client = MyFirstLinuxVM
 - Server = LinuxCentOS7



rsync – Remote Synchronization



rsync – Remote Synchronization

- **Basic syntax of rsync command**
 - `# rsync options source destination`
- **Install rsync in your Linux machine** (*check if it already exists*)
 - `# yum install rsync` (*On CentOS/Redhat based systems*)
 - `# apt-get install rsync` (*On Ubuntu/Debian based systems*)
- **rsync a file on a local machine**
 - `$ tar cvf backup.tar .` (*tar the entire home directory (/home/iafzal)*)
 - `$ mkdir /tmp/backups`
 - `$ rsync -zvh backup.tar /tmp/backups/`
- **rsync a directory on a local machine**
 - `$ rsync -azvh /home/iafzal /tmp/backups/`
- **rsync a file to a remote machine**
 - `$ mkdir /tmp/backups` (*create /tmp/backups dir on remote server*)
 - `$ rsync -avz backup.tar iafzal@192.168.1.x:/tmp/backups`
- **rsync a file from a remote machine**
 - `$ touch serverfile`
 - `$ rsync -avzh iafzal@192.168.1.x:/home/iafzal/serverfile /tmp/backups`



System Upgrade/Patch Management

- Two type of upgrades

Major version = 5, 6, 7

Minor version = 7.3 to 7.4

Major version = yum  command

Minor version = yum update 

Example:

`yum update -y`

yum update vs. upgrade

upgrade = delete packages



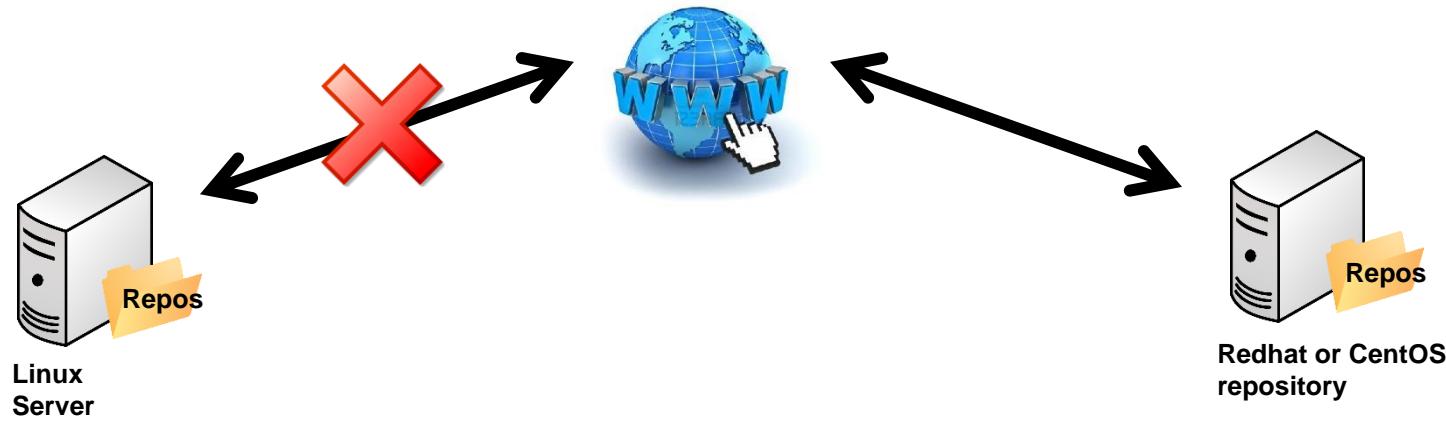
update = preserve



CREATE LOCAL REPOSITORY FROM DVD



- What is local repository?



- Command
createrepo

SSH AND TELNET

- Telnet = Un-secured connection between computers
- SSH = Secured
- Two type of packages for most of the services
 - Client package
 - Server package



SSH without a Password

- SSH is a secure way to login from host A to host B
- Repetitive tasks require login without a password



What we will learn...

- How to generate SSH keys on the server
- Add SSH keys to the client
- Verify by logging through SSH.

DNS = Domain Name System

- Purpose?

Hostname to IP	(A Record)
----------------	------------

IP to Hostname	(PTR Record)
----------------	--------------

Hostname to Hostname	(CNAME Record)
----------------------	----------------

- Files

`/etc/named.conf`

`/var/named`

- Service

`systemctl restart named`

Download, Install and Configure DNS

- Create a snapshot of your virtual machine
- Setup:
 - Master DNS
 - Secondary or Slave DNS
 - Client
- Domain Name = lab.local
- IP address = My local IP address on enp0s3
- Install DNS package
 - `yum install bind bind-utils -y`
- Configure DNS (**Summary**)
 - Modify `/etc/named.conf`
 - Create two zone files (`forward.lab` and `reverse.lab`)
 - Modify DNS file permissions and start the service
- Revert back to snapshot

HOSTNAME/IP LOOKUP

- Commands used for DNS lookup
 - **nslookup**
 - **dig**

NTP

- Purpose?

Time synchronization

- File

/etc/ntp.conf

- Service

systemctl restart ntpd

- Command

ntpq

chronyd

- Purpose? = Time synchronization
- Package name = chronyd
- Configuration file = /etc/chronyd.conf
- Log file = /var/log/chrony
- Service = systemctl start/restart chronyd
- Program command = chronyd.

New System Utility Command (`timedatectl`)

- The `timedatectl` command is a new utility for RHEL/CentOS 7/8 based distributions, which comes as a part of the systemd system and service manager
- It is a replacement for old traditional `date` command
- The `timedatectl` command shows/change date, time, and timezone
- It synchronizes the time with NTP server as well
 - You can either use `chrony` or `ntpd` and make the ntp setting in `timedatectl` as `yes`
 - Or you can use `systemd-timesyncd` daemon to synchronize time which is a replacement for ntpd and chrony

Please note:

Redhat/CentOS does not provide this daemon in its standard repo. You will have to download it separately.

New System Utility Command (`timedatectl`)

Lab exercise:

- To check time status
 - `timedatectl`
- To view all available time zones
 - `timedatectl list-timezones`
- To set the time zone
 - `timedatectl set-timezone "America/New_York"`
- To set date
 - `timedatectl set-time YYYY-MM-DD`
- To set date and time
 - `timedatectl set-time '2015-11-20 16:14:50'`
- To start automatic time synchronization with a remote NTP server
 - `timedatectl set-ntp true`.

Sendmail - OLD

- Purpose?

Send and receive emails

- Files

/etc/mail/sendmail.mc

/etc/mail/sendmail.cf

/etc/mail

- Service

systemctl restart sendmail

- Command

mail -s "subject line" email@mydomain.com

Sendmail

- **Sendmail** is a program in Linux operating systems that allows systems administrator to send email from the Linux system
- It uses SMTP (Simple Mail Transfer Protocol)
- SMTP port = 25
- It attempts to deliver the mail to the intended recipient immediately and, if the recipient is not present, it queues messages for later delivery.



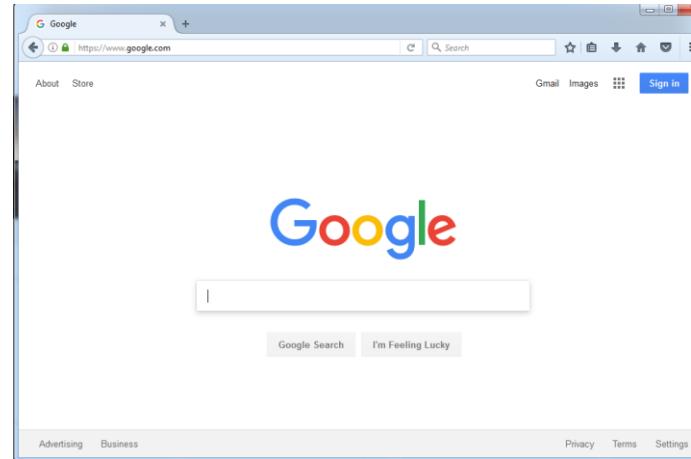
Sendmail

- Sendmail installation and configuration
 - `# su - (Login as root)`
 - `# rpm -qa | grep sendmail (verify if it is already installed)`
 - `# yum install sendmail sendmail-cf`
 - `# vi /etc/mail/sendmail.mc`
 - `# systemctl start sendmail`
 - `# systemctl enable sendmail`
 - `# systemctl stop firewalld`
 - `# systemctl disable firewalld`



Web Server (`httpd`)

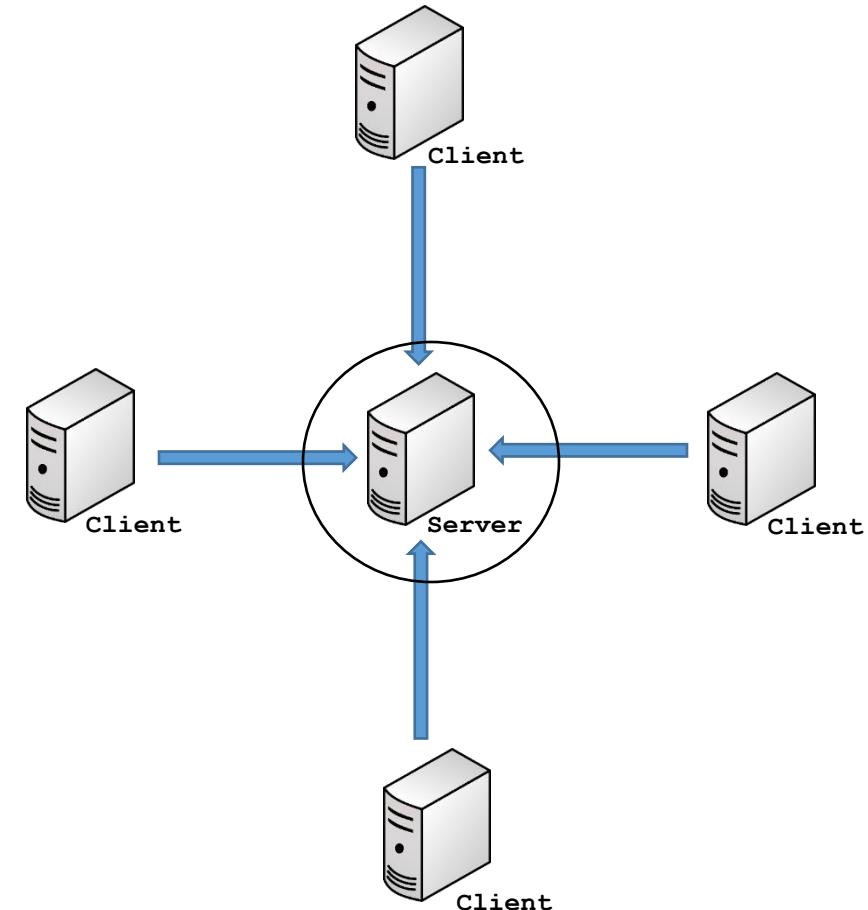
- Purpose = Serve webpages



- Service or Package name = `httpd`
- Files = `/etc/httpd/conf/httpd.conf`
`= /var/www/html/index.html`
- Log Files = `/var/log/httpd/`
- Service
 - `systemctl restart httpd`
 - `systemctl enable httpd`

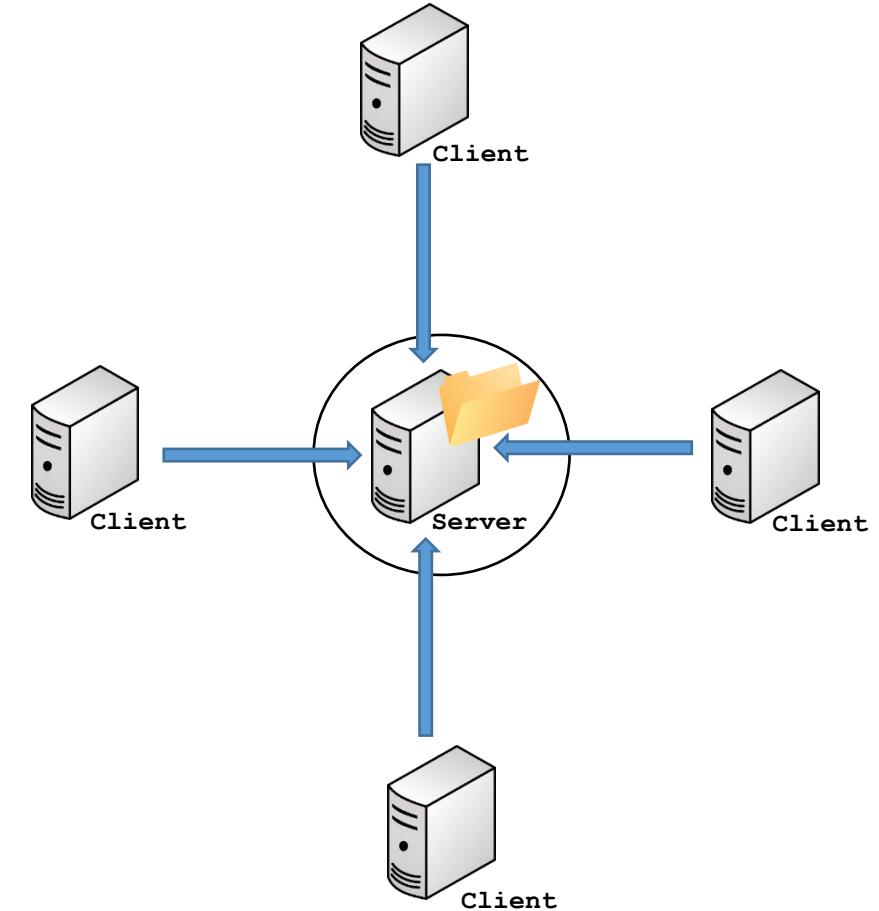
CENTRAL LOGGER (RSYSLOG)

- Purpose = Generate logs or collect logs from other servers
- Service or package name = **rsyslog**
- Configuration file= **/etc/syslog.conf**
- Service
 - `systemctl restart rsyslog`
 - `systemctl enable rsyslog`



NETWORK FILE SYSTEM (NFS)

- Purpose = Share files or directories (filesystem)
- Service or package name = **nfs-utils**
- Configuration file =
/etc/fstab, /etc(exports,
/etc/sysconfig/nfs
- Service
systemctl restart nfs-server
systemctl enable nfs-server



LINUX OS HARDENING



- User Account
- Remove un-wanted packages
- Stop un-used Services
- Check on Listening Ports
- Secure SSH Configuration
- Enable Firewall (iptables/firewalld)
- Enable SELinux
- Change Listening Services Port Numbers
- Keep your OS up to date (security patching)

OpenLDAP Installation

- What is OpenLDAP?
- OpenLDAP Service
 - slapd
- Start or stop the service
 - systemctl start slapd
 - systemctl enable slapd
- Configuration Files
 - /etc/openldap/slapd.d

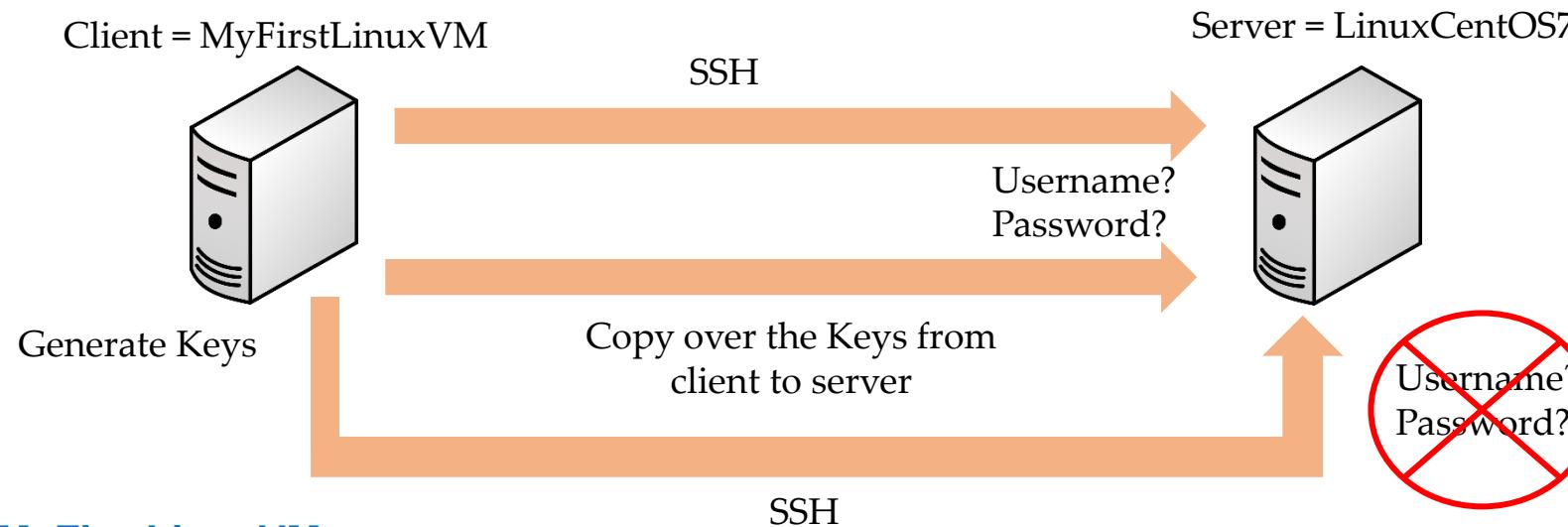
Trace Network Traffic (traceroute)

- The traceroute command is used in Linux to map the journey that a packet of information undertakes from its source to its destination. One use for traceroute is to locate when data loss occurs throughout a network, which could signify a node that's down.
 - Because each hop in the record reflects a new server or router between the originating PC and the intended target, reviewing the results of a traceroute scan also lets you identify slow points that may adversely affect your network traffic.
-
- Example
- ```
traceroute www.google.com
```

# Access Remote Server without Password (SSH-Keys)

- Two reasons to access a remote machine
  - Repetitive logins
  - Automation through scripts
- Keys are generated at user level
  - iafzal
  - root

# Access Remote Server without Password (SSH-Keys)



## Client = MyFirstLinuxVM

**Step 1** — Generate the Key

```
ssh-keygen
```

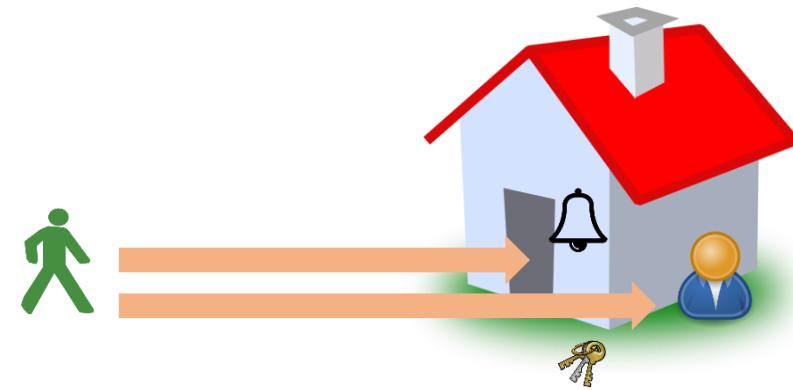
**Step 2** — Copy the Key to the server

```
ssh-copy-id root@192.168.1.x
```

**Step 3** — Login from client to server

```
ssh root@192.168.1.x
```

```
ssh -l root 192.168.1.x
```



# Cockpit

- Cockpit is a server administration tool sponsored by Red Hat, focused on providing a modern-looking and user-friendly interface to manage and administer servers
- Cockpit is the easy-to-use, integrated, glanceable, and open web-based interface for your servers
- The application is available in most of the Linux distributions such as, CentOS, Redhat, Ubuntu and Fedora
- It is installed in Redhat 8 by default and it is optional in version 7
- It can monitor system resources, add or remove accounts, monitor system usage, shut down the system and perform quite a few other tasks all through a very accessible web connection



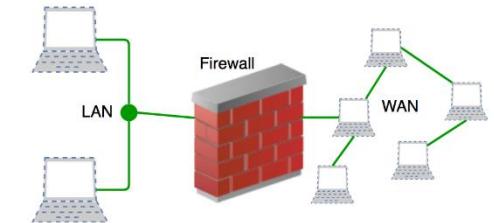
# Install, Configure and Manage Cockpit

- Check for network connectivity
  - `ping www.google.com`
- Install cockpit package as root
  - `yum/dnf install cockpit -y` (*For RH or CentOS*)
  - `apt-get install cockpit` (*For Ubuntu*)
- Start and enable the service
  - `systemctl start|enable cockpit`
- Check the status of the service
  - `systemctl status cockpit`
- Access the web-interface
  - `https://192.168.1.x:9090`

# Introduction to Firewall

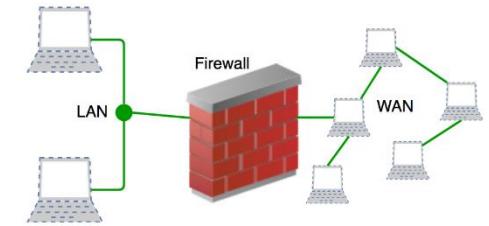
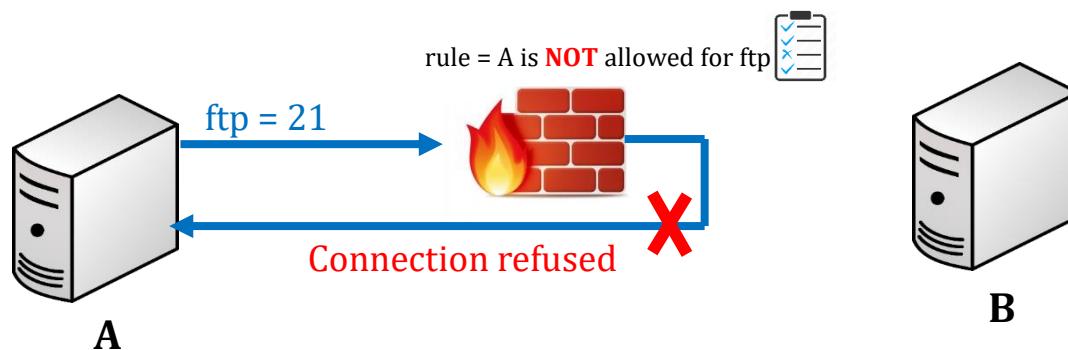
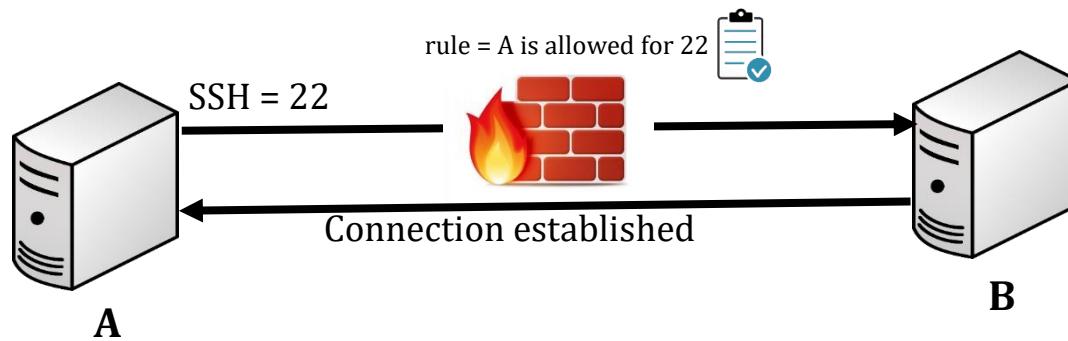
1/2

- What is Firewall
  - A wall that prevents the spread of fire
  - When data moves in and out of a server its packet information is tested against the firewall rules to see if it should be allowed or not
  - In simple words, a firewall is like a watchman, a bouncer, or a shield that has a set of rules given and based on that rule they decide who can enter and leave
- There are 2 type of firewalls in IT
  - Software = Runs on operating system
  - Hardware = A dedicated appliance with firewall software



# Introduction to Firewall

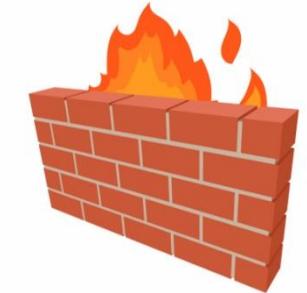
2/2



# Firewall (*firewalld*)

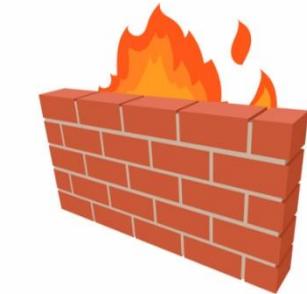
1/2

- Firewalld works the same way as iptables but of course it has its own commands
  - `firewall-cmd`
- It has a few pre-defined service rules that are very easy to turn on and off
  - Services such as: NFS, NTP, HTTPD etc.
- Firewalld also has the following:
  - Table
  - Chains
  - Rules
  - Targets



# Firewall (*firewalld*)

2/2



- You can run one or the other
  - iptables or firewalld
- Make sure iptables is stopped, disabled and mask
  - `systemctl stop iptables`
  - `systemctl disable iptables`
  - `systemctl mask iptables`
- Now check if firewalld package is installed
  - `rpm -qa | grep firewalld`
- Start firewalld
  - `systemctl start/enable firewalld`
- Check the rule of firewalld
  - `firewall-cmd --list-all`
- Get the listing of all services firewalld is aware of:
  - `firewall-cmd --get-services`
- To make firewalld re-read the configuration added
  - `firewall-cmd --reload`

# Firewall (`firewalld - Practical Examples`)

1/3

- The firewalld has multiple zone, to get a list of all zones
  - `firewall-cmd --get-zones`
- To get a list of active zones
  - `firewall-cmd --get-active-zones`
- To get firewall rules for public zone
  - `firewall-cmd --zone=public --list-all`  
OR
  - `firewall-cmd --list-all`
- All services are pre-defined by firewalld. What if you want to add a 3<sup>rd</sup> party service
  - `/usr/lib/firewalld/services/allservices.xml`
  - Simply cp any .xml file and change the service and port number

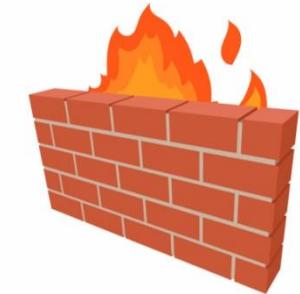


```
[root@MyFirstLinuxVM services]# cat test.xml
<?xml version="1.0" encoding="utf-8"?>
<service>
 <short>SSH</short>
 <description>To login</description>
 <port protocol="tcp" port="22"/>
</service>
[root@MyFirstLinuxVM services]#
```

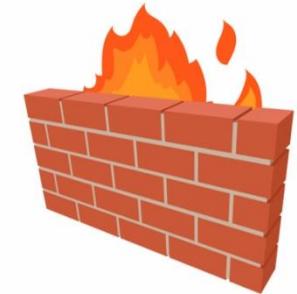
Diagram illustrating the XML structure of the test.xml file:

- Service**: Points to the `<service>` element.
- Port**: Points to the `<port protocol="tcp" port="22"/>` element.
- Description**: Points to the `<description>To login</description>` element.
- Version of XML**: Points to the XML declaration `<?xml version="1.0" encoding="utf-8"?>`.

- To add a service (http)
  - `firewall-cmd --add-service=http`
- To remove a service
  - `firewall-cmd --remove-service=http`
- To reload the firewalld configuration
  - `firewall-cmd --reload`
- To add or remove a service permanently
  - `firewall-cmd --add-service=http --permanent`
  - `firewall-cmd --remove-service=http --permanent`
- To add a service that is not pre-defined by firewalld
  - `/usr/lib/firewalld/services/allservices.xml`
  - Simply cp any .xml file sap.xml and change the service and port number (32)
  - `systemctl restart firewalld`
  - `firewall-cmd --get-services` (to verify new service)
  - `Firewall-cmd --add-service=sap`



- To add a port
  - `firewall-cmd --add-port=1110/tcp`
- To remove a port
  - `firewall-cmd --remove-port=1110/tcp`
- To reject incoming traffic from an IP address
  - `firewall-cmd --add-rich-rule='rule family="ipv4" source address="192.168.0.25" reject'`
- To block and unblock ICMP incoming traffic
  - `firewall-cmd --add-icmp-block-inversion`
  - `firewall-cmd --remove-icmp-block-inversion`
- To block outgoing traffic to a specific website/IP address
  - `host -t a www.facebook.com = find IP address`
  - `firewall-cmd --direct --add-rule ipv4 filter OUTPUT 0 -d 31.13.71.36 -j DROP`



# Tune System Performance

1/8

Linux system comes fine tuned by default when you install, however there are a few tweaks that can be done based on system performance and application requirements

In this lesson we will learn...

- Optimize system performance by selecting a tuning profile managed by the **tuned** daemon
- Prioritize or de-prioritize specific processes with the **nice** and **renice** commands

# Tune System Performance

2/8

## What is tuned?

- Tuned pronounced as tune-d
- Tune is for system tuning and d is for daemon
- It is **systemd** service that is used to tune Linux system performance
- It is installed in CentOS/Redhat version 7 and 8 by default
- **tuned** package name is **tuned**
- The **tuned** service comes with pre-defined profiles and settings (*List of profile will be discussed in the next page*)
- Based on selected profile the **tuned** service automatically adjust system to get the best performance. E.g. **tuned** will adjust networking if you are downloading a large file or it will adjust IO settings if it detects high storage read/write
- The tuned daemon applies system settings when the service starts or upon selection of a new tuning profile.

# Tune System Performance

(**tuned** profiles)

3/8

Tuned profile	Purpose
balanced	deal for systems that require a compromise between power saving and performance
desktop	Derived from the balanced profile. Provides faster response of interactive applications
Throughput-performance	Tunes the system for maximum throughput
Latency-performance	Ideal for server systems that require low latency at the expense of power consumption
network-latency	Derived from the latency-performance profile. It enables additional network tuning parameters to provide low network latency
Network-throughput	Derived from the throughput-performance profile. Additional network tuning parameters are applied for maximum network throughput
powersave	Tunes the system for maximum power saving
oracle	Optimized for Oracle database loads based on the throughput-performance profile
virtual-guest	Tunes the system for maximum performance if it runs on a virtual machine
virtual-host	Tunes the system for maximum performance if it acts as a host for virtual machines

# Tune System Performance

4 / 8

- Check if tuned package has been installed

```
rpm -qa | grep tuned
```

- Install tuned package if NOT installed already

```
yum install tuned
```

- Check **tuned** service status

```
systemctl status|enable|start tuned
```

```
systemctl enable tuned (To enable at boot time)
```

- Command to change setting for tuned daemon

```
tuned-adm
```

- To check which profile is active

```
tuned-adm active
```

- To list available profiles

```
tuned-adm list.
```

# Tune System Performance

5/8

- To change to desired profile

```
tuned-adm profile profile-name
```

- Check for tuned recommendation

```
tuned-adm recommend
```

- Turn off tuned setting daemon

```
tuned-adm off
```

- Change profile through web console

Login to <https://192.168.1.x:9090>

Overview → Configuration → Performance profile

# Tune System Performance

(**nice/renice**)

6/8

- Another way of keeping your system fine-tuned is by prioritizing processes through **nice** and **renice** command
- If a server has 1 CPU then it can execute **1** computation/process at a time as they come in (*first come first served*) while other processes must wait
- With **nice** and **renice** commands we can make the system to give preference to certain processes than others
- This priority can be set at 40 different levels
- The nice level values range from -20 (highest priority) to 19 (lowest priority) and by default, processes inherit their nice level from their parent, which is usually 0.

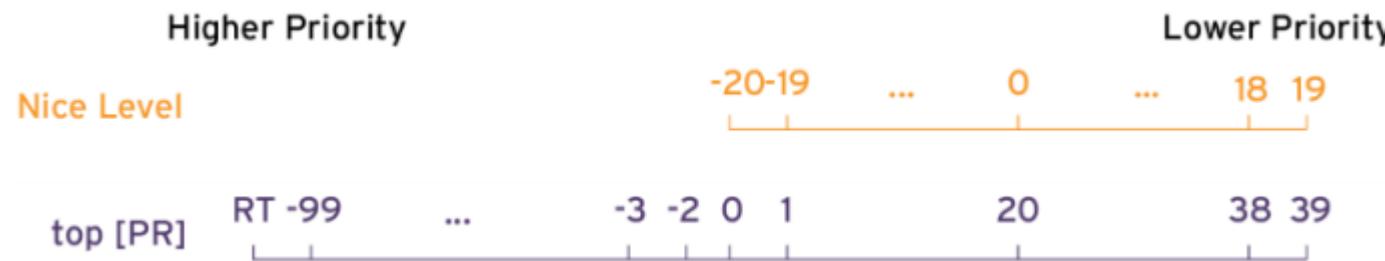
# Tune System Performance

(nice/renince)

7/8

- To check process priority

**top**



Nice value is a user-space and priority PR is the process's actual priority that use by Linux kernel. In Linux system priorities are 0 to 139 in which 0 to 99 for real time and 100 to 139 for users

- Process priority can be viewed through ps command as well with the right options

```
$ ps axo pid,comm,nice,cls --sort=-nice
```

# Tune System Performance

(nice/renice)

8/8

- To set the process priority

****nice -n # process-name****

**e.g. nice -n -15 top**

- To change the process priority

****renice -n # process-name****

**e.g. renice -n 12 PID.**

# Run Containers

## What is a Container?

- The term container and the concept came from the shipping container



- These containers are shipped from city to city and country to country
- No matter which part of the world you go to, you will find these containers with the exact same measurements... **YOU KNOW WHY???**
- Because around the world all docks, trucks, ships and warehouses are built to easily transport and store them



# Run Containers

## What is a Container?

Now when we are talking about containers in IT we are fulfilling somewhat similar purpose

### Please Note:

**Container technology is mostly used by developers or programmers  
who write codes to build applications**

**As a system administrator your job is to install, configure and  
manage them.**

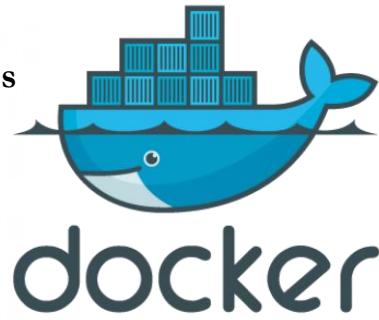
You can move the application anywhere without moving its OS just like moving the actual physical container anywhere that would fit on any dockyard, truck, ship or warehouse

- An OS can run single or multiple containers at the same time

# Run Containers

## What are the Container Software?

Developed by:  
Solomon Hykes



Released on:  
March 20<sup>th</sup> 2013

Developed by:  
 Red Hat



The word "podman" is written in a large, lowercase, sans-serif font. The letters "pod" are in a light purple color, while "man" is in black.

- Docker is the software used to create and manage containers
- Just like any other package, docker can be installed on your Linux system and its service or daemon can be controlled through native Linux service management tool

- Podman is an alternative to docker
- Docker is not supported in RHEL 8
- It is daemon less, open source, Linux-native tool designed to develop, manage, and run containers.

# Run Containers

## Getting Familiar with Redhat Container Technology

Red Hat provides a set of command-line tools that can operate without a container engine, these include:

- **podman** - for directly managing pods and container images (run, stop, start, ps, attach, etc.)
- **buildah** - for building, pushing, and signing container images
- **skopeo** - for copying, inspecting, deleting, and signing images
- **runc** - for providing container run and build features to podman and buildah
- **crun** - an optional runtime that can be configured and gives greater flexibility, control, and security for rootless containers.

## Getting Familiar with podman Container Technology

When you hear about containers then you should know the following terms as well

- **images** – containers can be created through images and containers can be converted to images
- **pods** – Group of containers deployed together on the host. In the podman logo there are 3 seals grouped together as a pod.



podman

# Run Containers

## Building, Running and Managing Containers



To install podman

- `yum/dnf install podman -y`
- `yum install docker -y` (*For dockers*)

Creating alias to docker

- `alias docker=podman`

Check podman version

- `podman -v`

Getting help

- `podman --help or man podman`

Check podman environment and registry/repository information

- `podman info` (*If you are trying to load a container image, then it will look at the local machine and then go through each registry by the order listed*)

To search a specific image in repository.

- `podman search httpd`

# Run Containers

## Building, Running and Managing Containers



To list any previously downloaded podman images

- **podman images**

To download available images

- **podman pull docker.io/library/httpd**
- **podman images** (*Check downloaded image status*)

To list podman running containers

- **podman ps**

To run a downloaded httpd containers

- **podman run -dt -p 8080:80/tcp docker.io/library/httpd**  
*(d=detach, t=get the tty shell, p=port)*
- **podman ps**      or *Check httpd through web browser*

To view podman logs.

- **podman logs -l**

# Run Containers

## Building, Running and Managing Containers



To stop a running container

- **podman stop con-name**    (*con-name from podman ps command*)
- **podman ps**                (*To list running containers*)

To run a multiple containers of httpd by changing the port #

- **podman run -dt -p 8081:80/tcp docker.io/library/httpd**
- **podman run -dt -p 8082:80/tcp docker.io/library/httpd**
- **podman ps**

To stop and start a previously running container

- **podman stop|start con-name**

To create a new container from the downloaded image

- **podman create --name httpd-con docker.io/library/httpd**

To start the newly created container.

- **podman start httpd-con**

# Run Containers

## Building, Running and Managing Containers



### Manage containers through systemd

- First you have to generate a unit file
  - `podman generate systemd --new --files --name httpd-con`
- Copy it systemd directory
  - `cp /root/container-httpd.service /etc/systemd/system`
- Enable the service
  - `systemctl enable container-httpd-con.service`
- Start the service.
  - `systemctl start container-httpd-con.service`

## **Network:**

### **What is a network Interface?**

A network interface is the point of interconnection between a computer and a private or public network. A network interface is generally a physical network interface card (NIC)

A network interface card (NIC) is a circuit board or card that is installed in a computer so that it can be connected to a network.

A network interface card provides the computer with a dedicated, full-time connection to a network. Personal computers and workstations on network typically contain a network interface card specifically designed for the LAN transmission technology

### **What is an IP address?**

The IP stands for internet protocol address

An Internet Protocol address is a numerical label assigned to each device participating in a computer network that uses the Internet Protocol for communication. [Wikipedia](#)

In simple word a computer needs and IP address to connect to another computer and the other computer must also have an IP to listen to our communication

### **What is Subnet mask?**

When you configure the TCP/IP protocol, an IP address, **subnet mask**, and usually a default gateway are required in the TCP/IP configuration settings. To configure TCP/IP correctly, it is necessary to understand how networks are addressed and divided into networks and subnetworks.

A complete article on subnet mask can be viewed from the following link:

<http://www.iplocation.net/tools/netmask.php>

### **What is Network Gateway?**

In computer networking, a gateway is a node (a router) on a TCP/IP network that serves as an access point to another network. A default gateway is the node on the computer network that the network software uses when an IP address does not match any other routes in the routing table.

### **What is MAC address?**

The MAC address is a unique value associated with a network adapter. MAC addresses are also known as hardware addresses or physical addresses. They uniquely identify an adapter on a LAN. MAC addresses are 12-digit hexadecimal numbers (48 bits in length). By convention, MAC addresses are usually written in one of the following two formats:

MM:MM:MM:SS:SS:SS

### **What is Static and DHCP IP?**

Static as it sounds stays permanent where DHCP IP is a floating IP and changes every time the system reboots. For more information on difference between Status and DHCP visit the following link:

<http://www.differencebetween.net/technology/difference-between-dhcp-and-static-ip/>

## **LAN, MAN and WAN?**

Local Area Network = Computer connected with one physical location

Metropolitan Area Network = Computers connected with one region to another (or state to state)

Wide Area Network = Country to country

## **Interface Status or Configuration:**

To display current interface MAC address

`ifconfig -a`

To display current interfaces

`ifconfig -a`

To assign an IP address to interfaces

`ifconfig eth0 192.168.1.2`

To bring an interface down

`ifconfig eth0 down`

To bring an interface down

`ifconfig eth0 up`

## **Network Configuration Files:**

### **Interface, Netmask and defaultgateway files**

`/etc/sysconfig/network-scripts/ifcfg-eth0` (Linux)

`/etc/hosts`

`/etc/modprobe.conf` (Linux)

`/etc/resolv.conf` (Both OS)

`/etc/nsswitch.conf` (Both OS)

## **Network Commands**

`ping` = ping command allows to check the network status of another computer

`ifconfig` = This command allows you to check or assign IP address to an interface

`netstat` = Print network connections, routing tables, interface statistics, gateway information etc.

`tcpdump` = To view the incoming and outgoing on the system (`tcpdump -i eth0`)

## **Routes Setup:**

### **Linux Machine**

Adding a route

`/sbin/route add -net 192.168.1.2 netmask 255.255.255.0 gw 192.168.1.1`

Deleting a route

```
/sbin/route del -net 192.168.1.2 netmask 255.255.255.0 gw 192.168.1.1
```

Add a new interface

```
/sbin/ifconfig interface plumb
```

Bring up the new interface

```
/sbin/ifconfig interface up
```

Configure a new interface

```
/sbin/ifconfig interface 192.168.1.2
```

## Network Related Utilities

### ping

- The ping command sends an echo request to a network host. It is useful for:
  - Determining the status of the network and various foreign hosts.
  - Tracking and isolating hardware and software problems.
  - Testing, measuring, and managing networks.
- The ping command sends one datagram per second and prints one line of output for every response received. Round-trip times and packet loss statistics are calculated and displayed.  
Example:

```
% ping kanaha or 164.122.27.33
PING kanaha.mhpcc.edu: (164.122.27.33): 56 data bytes
64 bytes from 164.122.27.33: icmp_seq=0 ttl=254 time=3 ms
64 bytes from 164.122.27.33: icmp_seq=1 ttl=254 time=2 ms
64 bytes from 164.122.27.33: icmp_seq=2 ttl=254 time=2 ms
64 bytes from 164.122.27.33: icmp_seq=3 ttl=254 time=2 ms
^C
----kanaha.mhpcc.edu PING Statistics----
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 2/2/3 ms
```

### traceroute

- The traceroute command prints the route that IP packets take to a network host. It is intended for use in network testing, measurement, and management.  
Example:

```
% traceroute archie.rutgers.edu
traceroute to dorm.Rutgers.EDU (128.6.18.15), 30 hops max, 40 byte
packets
 1 B2_IGSL_01 (129.24.96.1) 2 ms 2 ms 2 ms
 2 FZ00_rtr_01 (129.24.56.1) 3 ms 2 ms 7 ms
 3 msh (129.24.8.193) 5 ms 7 ms 4 ms
 4 198.83.5.5 (198.83.5.5) 7 ms 4 ms 7 ms
 5 hssi3-0.cnss116.Albuquerque.t3.ans.net (192.103.74.41) 5 ms 4 ms 6
ms
 6 mf-0.cnss112.Albuquerque.t3.ans.net (140.222.112.222) 4 ms 4 ms 4
ms
 7 t3-0.cnss64.Houston.t3.ans.net (140.222.64.1) 30 ms 30 ms 30 ms
 8 t3-0.cnss80.St-Louis.t3.ans.net (140.222.80.1) 47 ms 47 ms 46 ms
 9 t3-1.cnss25.Chicago.t3.ans.net (140.222.25.2) 54 ms 52 ms 53 ms
10 t3-0.cnss40.Cleveland.t3.ans.net (140.222.40.1) 60 ms 59 ms 59
ms
11 t3-1.cnss48.Hartford.t3.ans.net (140.222.48.2) 73 ms 78 ms 74 ms
12 t3-2.cnss32.New-York.t3.ans.net (140.222.32.3) 78 ms 76 ms 76 ms
13 t3-0.enss137.t3.ans.net (140.222.137.1) 79 ms 80 ms 86 ms
14 fenchurch-gateway.jvnc.net (192.12.211.65) 83 ms 80 ms 84 ms
15 airport2-gateway.jvnc.net (130.94.9.250) 84 ms 86 ms 88 ms
16 airport1-gateway.jvnc.net (130.94.7.1) 85 ms 92 ms 84 ms
17 rutgers-gateway.jvnc.net (130.94.7.10) 89 ms 86 ms 90 ms
18 rucs-gw.rutgers.edu (128.6.21.7) 94 ms 104 ms 95 ms
```

```
19 dorm.rutgers.edu (128.6.18.15) 92 ms 93 ms 91 ms
```

- Warning: Because of the load traceroute imposes on the network, the traceroute command should not be used during normal operations or from automated scripts.
- The traceroute utility may not be available on all systems.

## ftp

- ftp stands for File Transfer Protocol. File transfer provides a means for you to obtain computer files (text, image, sound, etc.) from other computers over the network.
- ftp can also be used to send (upload) files from your computer to another computer, providing you have write permission or a real account on the machine you are uploading.
- The ftp utility has its own set of UNIX like commands which allow you to perform tasks such as:
  - Connect and login to a remote host
  - Navigate directories
  - List directory contents
  - Put and get files
  - Transfer files as ascii, ebcDIC or binary
- A sample ftp session appears below. The commands which are entered by the user are in bold type.

```
kanaha% ftp makena.mhpcc.edu
Connected to makena.mhpcc.edu.
220 makena.mhpcc.edu FTP server (Version 4.9 Thu Sep 2 20:35:07 CDT
1993)
Name (makena.mhpcc.edu:jsmith): jsmith
331 Password required for jsmith.
Password:
230 User jsmith logged in.
ftp> dir
200 PORT command successful.
150 Opening data connection for /bin/ls.
total 1464
drwxr-sr-x 3 jsmith staff 1024 Mar 11 20:04 Mail
drwxr-sr-x 2 jsmith staff 1536 Mar 3 18:07 Misc
drwxr-sr-x 5 jsmith staff 512 Dec 7 10:59 OldStuff
drwxr-sr-x 2 jsmith staff 1024 Mar 11 15:24 bin
drwxr-sr-x 5 jsmith staff 3072 Mar 13 16:10 mpl
-rw-r--r-- 1 jsmith staff 209671 Mar 15 10:57 myfile.out
drwxr-sr-x 3 jsmith staff 512 Jan 5 13:32 public
drwxr-sr-x 3 jsmith staff 512 Feb 10 10:17 pvm3
226 Transfer complete.
ftp> cd mpl
250 CWD command successful.
ftp> dir
200 PORT command successful.
150 Opening data connection for /bin/ls.
total 7320
-rw-r--r-- 1 jsmith staff 1630 Aug 8 1994 dboard.f
-rw-r----- 1 jsmith staff 4340 Jul 17 1994 vttest.c
-rwxr-xr-x 1 jsmith staff 525574 Feb 15 11:52 wave_shift
-rw-r--r-- 1 jsmith staff 1648 Aug 5 1994 wide.list
-rwxr-xr-x 1 jsmith staff 4019 Feb 14 16:26 fix.c
```

```
226 Transfer complete.
ftp> get wave_shift
200 PORT command successful.
150 Opening data connection for wave_shift (525574 bytes).
226 Transfer complete.
528454 bytes received in 1.296 seconds (398.1 Kbytes/s)
ftp> quit
221 Goodbye.
```

- Many computers on the Internet permit *anonymous ftp*. You can login to these machines without a real account, to obtain files which have been made publicly available. Typically, the user name **anonymous** is used, coupled with your email address as the password.
- Anonymous ftp is usually restricted so that users can only see what the server permits them to see. Anonymous users do not have full privileges as would a user with a real computer account.

### **telnet**

- Telnet is a utility that allows a computer user at one site to make a connection, login and then conduct work on a computer at another site. For example, you can use the telnet command to run a program in your directory on a supercomputer thousands of miles away.
  - Telnet is used to access many of the Internet resources, such as databases, libraries and computers
- Example telnet session:

```
% telnet makena
Trying...
Connected to makena.mhpcc.edu.
Escape character is '^]'.

AIX Version 3
(C) Copyrights by IBM and by others 1982, 1993.
login: jsmith
jsmith's Password:

*
*
*
*
* WELCOME TO THE Maui High Performance Computing Center *
*
*
*
*
*

Last unsuccessful login: Fri Mar 3 12:01:09 HST 1995 on pts/0 from
kanaha.mhpcc.edu
Last login: Wed Mar 8 18:33:27 HST 1995 on pts/10

{ do some work }
```

```
makena% logout
Connection closed.
```

## rlogin

### rsh

### rcp

- rlogin (remote login), rsh (remote shell) and rcp (remote copy) are three utilities which allow you to perform tasks on other machines without requiring the usual login authentication.
- All three utilities depend upon a *.rhosts* located in your home directory. The *.rhosts* file contains the names of your "trusted" hosts and your userid on each of those hosts. An example appears below:

```
apache.unm.edu jsmith
zeus.mit.edu jsmith
athena.com smith
fox.eeco.org smithj
```

- rlogin: Allows you to login to a remote machine. It is nearly identical to telnet in function and appearance, however if your *.rhosts* file is setup accordingly, you will be able to login to your account on another machine without having to enter a userid and password.
- rsh: The remote shell command can be used to execute a command on remote host or log into remote host. With the proper *.rhosts* file, authentication is not required.

Examples:

<b>rsh host2</b>	- <i>will connect to host2 for login</i>
<b>rsh host2 df</b>	- <i>check the amount of free disk space on remote host2</i>
<b>rsh host2 ps aux  grep jsmith</b>	- <i>check for processes owned by jsmith on host2</i>
<b>rsh host2 rm /tmp/myfile.old</b>	- <i>remove a file in host2</i>
<b>rsh host2 cat test1 "&gt;&gt;" test2</b>	- <i>append test1 file on remote host to test2 file on remote host</i>
<b>rsh host2 cat test1 &gt;&gt; test2</b>	- <i>append test1 file on remote host to test2 file on local host</i>

- rcp: Remote copy enables you to copy files between different systems. With the proper *.rhosts* file, authentication is not required.

Example:

```
rcp localfile host2:/home/eng/journal
```

## DNS (Domain Name Server)

The DNS translates Internet domain and host names to IP addresses. DNS automatically converts the names we type in our Web browser address bar to the IP addresses of Web servers hosting those sites.

DNS implements a distributed database to store this name and address information for all public hosts on the Internet. DNS assumes IP addresses do not change (are statically assigned rather than dynamically assigned).

```
Hostname to 192.168.1.2 (Called A Record)
192.168.1.2 to hostname (Called PTR Record)
Hostname to hostname (Called CNAME Record)
```

In a DNS server such as BIND (Berkeley Internet Name Domain), all information is stored in basic data elements called resource records (RR). The resource record is usually a fully qualified domain name (FQDN) of a host, and is broken down into multiple sections organized into a tree-like hierarchy. This hierarchy consists of a main trunk, primary branches, secondary branches, and so on.

BIND consists of a set of DNS-related programs. It contains a nameserver called named, an administration utility called rndc, and a debugging tool called dig

### How does DNS resolution work?

A client application requests an IP address from the name server usually by connecting to UDP port 53. The name server will attempt to resolve the FQDN based on its resolver library, which may contain authoritative information about the host requested or cached data about that name from an earlier query. If the name server does not already have the answer, it will turn to root name servers to determine the authoritative for the FQDN in question. Then, with that information, it will query the authoritative name servers for that name to determine the IP address.

### What is an MX record?

An MX record numerically ranks the mail servers you would prefer to receive email for a domain. The MX record with the lowest number is preferred over the others, but you can set multiple email servers with the same value for simple load balancing.

When the named service is started, it reads the configuration from files as described in the following table

Path	Description
/etc/named.conf	The main configuration file.
/etc/named/	An auxiliary directory for configuration files that are included in the main configuration file

The process name of the service	=	named
Name of the rpm	=	bind

# Install and Configure DNS

## System Information

Domain Name = lab.local  
IP address = My local IP address on enp0s3

### **Step 1. Install DNS package**

```
yum install bind bind-utils -y
```

### **Step 2. Configure DNS Server**

```
vi /etc/named.conf
```

#### Edit the line

```
listen-on port 53 { 127.0.0.1; };
with
listen-on port 53 { 127.0.0.1; 192.168.1.29; };
```

Go to the bottom of the file before "include" line and add

```
zone "lab.local" IN {
 type master;
 file "forward.lab";
 allow-update { none; };
};

zone "1.168.192.in-addr.arpa" IN {
 type master;
 file "reverse.lab";
 allow-update { none; };
};

include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
```

Save and quit

### **Step 3. Create Zone Files**

```
cd /var/named
touch forward.lab
touch reverse.lab
```

#### **Step 4: Modify the newly created Zone files – Forward zone file**

Add the following lines:

```
$TTL 86400
@ IN SOA masterdns.lab.local. root.lab.local. (
 2011071001 ;Serial
 3600 ;Refresh
 1800 ;Retry
 604800 ;Expire
 86400 ;Minimum TTL
)
@ IN NS masterdns.lab.local.
@ IN A 192.168.1.29
masterdns IN A 192.168.1.29
clienta IN A 192.168.1.240
clientb IN A 192.168.1.241
```

#### **Step 5: Modify the newly created Zone files – Reverse zone file**

Add the following lines:

```
$TTL 86400
@ IN SOA masterdns.lab.local. root.lab.local. (
 2011071001 ;Serial
 3600 ;Refresh
 1800 ;Retry
 604800 ;Expire
 86400 ;Minimum TTL
)
@ IN NS masterdns.lab.local.
@ IN PTR lab.local.
masterdns IN A 192.168.1.29
158 IN PTR masterdns.lab.local.
240 IN PTR clienta.lab.local.
241 IN PTR clientb.lab.local.
```

#### **Step 6: Start the DNS server**

```
systemctl start named
systemctl enable named
```

**Step 7: Disable firewalld**

```
systemctl stop firewalld
systemctl disable firewalld
```

**Step 8. Configuring Permissions, Ownership, and SELinux**

```
chgrp named -R /var/named
chown -v root:named /etc/named.conf
restorecon -rv /var/named
restorecon /etc/named.conf
```

**Step 9. Test DNS configuration and zone files for any syntax errors**

```
named-checkconf /etc/named.conf
named-checkzone lab.local /var/named/forward.lab
named-checkzone lab.local /var/named/reverse.lab
```

**Step 10. Add DNS Server Information to network file**

```
vi /etc/sysconfig/network-script/ifcfg-enp0s3
```

**DNS=192.168.1.29**

**Step 11. Modify /etc/resolv.conf**

```
nameserver 192.168.1.29
```

**Step 12. Restart network service**

```
systemctl restart network
```

**Step 13. Test DNS server**

```
dig masterdns.lab.local
nslookup masterdns.lab.local
nslookup clienta.lab.local
nslookup clientb.lab.local
nslookup 192.168.1.240
nslookup 192.168.1.241
```

## Using the rndc Utility

The rndc utility is a command line tool that allows you to administer the named service, both locally and from a remote machine. Its usage is as follows:

```
rndc [option...] command [command-option]
```

The rndc configuration is located in /etc/rndc.conf. If the file does not exist, the utility will use the key located in /etc/rndc.key, which was generated automatically during the installation process using the rndc-confgen -a command.

### Checking the Service Status

To check the current status of the named service, use the following command:

```
rndc status
```

To reload both the configuration file and zones, type the following at a shell prompt:

```
rndc reload
```

*This will reload the zones while keeping all previously cached responses, so that you can make changes to the zone files without losing all stored name resolutions.*

To reload a single zone, specify its name after the reload command, for example:

```
rndc reload localhost
```

Finally, to reload the configuration file and newly added zones only, type:

```
rndc reconfig
```

## Zone File Parameters Information:

### \$TTL

The \$TTL directive allows you to set the default Time to Live (TTL) value for the zone, that is, how long is a zone record valid. Each resource record can contain its own TTL value, which overrides this directive.

Increasing this value allows remote nameservers to cache the zone information for a longer period of time, reducing the number of queries for the zone and lengthening the amount of time required to propagate resource record changes.

Example: Using the \$TTL directive

```
$TTL 1D
```

### **Common Resource Records**

The following resource records are commonly used in zone files:

A

The Address record specifies an IP address to be assigned to a name. It takes the following form:

```
hostname IN A IP-address
```

If the hostname value is omitted, the record will point to the last specified hostname.

“Using the A resource record”, the requests for server1.example.com are pointed to 10.0.1.3 or 10.0.1.5.

Using the A resource record

```
server1 IN A 10.0.1.3 IN A 10.0.1.5
```

### **CNAME**

The Canonical Name record maps one name to another. Because of this, this type of record is sometimes referred to as an alias record. It takes the following form:

```
alias-name IN CNAME real-name
```

CNAME records are most commonly used to point to services that use a common naming scheme, such as www for Web servers. However, there are multiple restrictions for their usage:

CNAME records should not point to other CNAME records. This is mainly to avoid possible infinite loops. CNAME records should not contain other resource record types (such as A, NS, MX, etc.). The only exception are DNSSEC related records (that is, RRSIG, NSEC, etc.) when the zone is signed. Other resource record that point to the fully qualified domain name (FQDN) of a host (that is, NS, MX, PTR) should not point to a CNAME record.

“Using the CNAME resource record”, the A record binds a hostname to an IP address, while the CNAME record points the commonly used www hostname to it.

```
www IN CNAME server1
```

### **MX**

The Mail Exchange record specifies where the mail sent to a particular namespace controlled by this zone should go. It takes the following form:

```
IN MX preference-value email-server-name
```

The email-server-name is a fully qualified domain name (FQDN). The preference-value allows numerical ranking of the email servers for a namespace, giving preference to some email systems over others. The MX resource record with the lowest preference-value is preferred over the others. However, multiple email servers can possess the same value to distribute email traffic evenly among them.

“Using the MX resource record”, the first mail.example.com email server is preferred to the mail2.example.com email server when receiving email destined for the example.com domain.  
`example.com. IN MX 10 mail.example.com. IN MX 20 mail2.example.com`

## NS

The Nameserver record announces authoritative nameservers for a particular zone. It takes the following form:

`IN NS nameserver-name`

The nameserver-name should be a fully qualified domain name (FQDN). Note that when two nameservers are listed as authoritative for the domain, it is not important whether these nameservers are secondary nameservers, or if one of them is a primary server. They are both still considered authoritative.

Using the NS resource record

`IN NS dns1.example.com.  
IN NS dns2.example.com.`

## PTR

The Pointer record points to another part of the namespace. It takes the following form:

`last-IP-digit IN PTR FQDN-of-system`

The last-IP-digit directive is the last number in an IP address, and the FQDN-of-system is a fully qualified domain name (FQDN).

PTR records are primarily used for reverse name resolution, as they point IP addresses back to a particular name. Refer to Section 15.2.2.4.2, “A Reverse Name Resolution Zone File” for more examples of PTR records in use

## SOA

The Start of Authority record announces important authoritative information about a namespace to the nameserver. Located after the directives, it is the first resource record in a zone file. It takes the following form:

```
@ IN SOA primary-name-server hostmaster-email (
 serial-number
 time-to-refresh
 time-to-retry
 time-to-expire
 minimum-TTL)
```

The directives are as follows:

- The @ symbol places the \$ORIGIN directive (or the zone's name if the \$ORIGIN directive is not set) as the namespace being defined by this SOA resource record.

- The primary-name-server directive is the hostname of the primary nameserver that is authoritative for this domain.
- The hostmaster-email directive is the email of the person to contact about the namespace.
- The serial-number directive is a numerical value incremented every time the zone file is altered to indicate it is time for the named service to reload the zone.
- The time-to-refresh directive is the numerical value secondary nameservers use to determine how long to wait before asking the primary nameserver if any changes have been made to the zone.
- The time-to-retry directive is a numerical value used by secondary nameservers to determine the length of time to wait before issuing a refresh request in the event that the primary nameserver is not answering. If the primary server has not replied to a refresh request before the amount of time specified in the time-to-expire directive elapses, the secondary servers stop responding as an authority for requests concerning that namespace.
- In BIND 4 and 8, the minimum-TTL directive is the amount of time other nameservers cache the zone's information. In BIND 9, it defines how long negative answers are cached for. Caching of negative answers can be set to a maximum of 3 hours (that is, 3H).

When configuring BIND, all times are specified in seconds. However, it is possible to use abbreviations when specifying units of time other than seconds, such as minutes (M), hours (H), days (D), and weeks (W). Following table, “Seconds compared to other time units” shows an amount of time in seconds and the equivalent time in another format.

Seconds compared to other time units

Seconds	Other Time Units
60	1M
1800	30M
3600	1H
10800	3H
21600	6H
43200	12H
86400	1D
259200	3D
604800	1W
31536000	365D

Example: Using the SOA resource record

```
@ IN SOA dns1.example.com. hostmaster.example.com. (
2001062501 ; serial
21600 ; refresh after 6 hours
3600 ; retry after 1 hour
604800 ; expire after 1 week
86400) ; minimum TTL of 1 day
```

### A simple zone file:

```
$ORIGIN example.com.
$TTL 86400
@ IN SOA dns1.example.com. hostmaster.example.com. (
```

```

2001062501 ; serial
21600 ; refresh after 6 hours
3600 ; retry after 1 hour
604800 ; expire after 1 week
86400) ; minimum TTL of 1 day

; Comments
; Comments

 IN NS dns1.example.com.
 IN NS dns2.example.com.

dns1 IN A 10.0.1.1
dns2 IN A 10.0.1.2
dns1 IN CNAME server1

```

## Some Important Questions Regarding DNS

**Q:1 What does BIND Stands for ?**

**Ans:** BIND stands for Berkeley Internet Name Domain.

**Q:2 What is DNS Server and its fundamentals ?**

**Ans:** The Domain Name System (DNS) is a hierarchical, distributed database. It stores information for mapping Internet host names to IP addresses and vice versa, mail routing information, and other data used by Internet applications. Clients look up information in the DNS by calling a resolver library, which sends queries to one or more name servers and interprets the responses. The BIND 9 software distribution contains a name server, named, and a resolver library, liblwres.

**Q:3 What is the default port of BIND ?**

**Ans:** The BIND server is accessed via the network on port 53. Both TCP and UDP ports are used. Queries are made via UDP & Responses are made via UDP unless the response is too large to fit in a single packet , If the response won't fit in a single UDP packet, then the response is returned via TCP.

**Q:4 How will you define Domain Name ?**

**Ans:** The data stored in the DNS is identified by domain names that are organized as a tree according to organizational or administrative boundaries. Each node of the tree, called a domain, is given a label. The domain name of the node is the concatenation of all the labels on the path from the node to the root node. This is represented in written form as a string of labels listed from right to left and separated by dots. A label need only be unique within its parent domain.

For example, a domain name for a host at the company Linuxtech, Inc. could be mail.linuxtech.com, where com is the top level domain to which mail.linuxtech.com belongs, example is a subdomain of com, and 'mail' is the name of the host

**Q:5 What are zone files in DNS server ?**

Ans: The files which contain the data being served by the DNS system are called “Zone Files” They are made up of a series of “Resource Records”. A Zone File will always contain an SOA record as well as additional records.

Q:6 What are the different types of DNS Server ?

Ans: Primary Master : The authoritative server where the master copy of the zone data is maintained is called the primary master server, or simply the primary. Typically it loads the zone contents from some local file edited by humans or perhaps generated mechanically from some other local file which is edited by humans. This file is called the zone file or master file.

Slave Server : The other authoritative servers, the slave servers (also known as secondary servers) load the zone contents from another server using a replication process known as a zone transfer. Typically the data are transferred directly from the primary master, but it is also possible to transfer it from another slave. In other words, a slave server may itself act as a master to a subordinate slave server.

Caching Name Server : Caching Name server is not authoritative for any zone, all queries are forwarded to other DNS servers if they are not stored in the DNS-cache zone. Answers for all queries are cached in DNS-cache zone for a time.

Forwarding : In this type of DNS server , all queries are forwarded to a specific list of name servers

Q:7 How the load balancing is achieved using DNS ?

Ans: A primitive form of load balancing can be achieved in the DNS by using multiple records (such as multiple A records) for one name. For example, if you have three WWW servers with network addresses of 10.0.0.1, 10.0.0.2 and 10.0.0.3, a set of records such as the following means that clients will connect to each machine one third of the time

multiple-a-records

When a resolver queries for these records, BIND will rotate them and respond to the query with the records in a different order. In the example above, clients will randomly receive records in the order 1,2, 3; 2, 3, 1; and 3, 1, 2. Most clients will use the first record returned and discard the rest.

Q:8 How to check syntax of named.conf is correct or not ?

Ans: named-checkconf is the command, which checks the syntax of named.conf file.

```
named-checkconf /etc/named.conf
```

If bind is running in chroot environment use below command

```
named-checkconf -t /var/named/chroot /etc/named.conf
```

Q:9 What are the different types of Resource Records in bind ?

Ans: Below are the list of resource records in bind :

SOA – start of authority, for a given zone

NS – name server

A – name-to-address mapping

PTR – address-to-name mapping

CNAME – canonical name (for aliases)

MX – mail exchanger (host to receive mail for this name)

TXT – textual info

RP – contact person for this zone

WKS – well known services

HINFO – host information

Comments start with ; continue to end of line

Q:10 Explain Bind chroot environment ?

Ans: Running bind in a chroot environment means named process will be limited to their directory only (/var/named/chroot). This can help improve system security by placing BIND in a "sandbox", which will limit the damage done if a server is compromised.

Q:11 What is domain delegation in Bind ?

Ans: Domain delegation means fully delegate the responsibility for a sub-domain to another name server.

Exmaple :

```
squid.linuxtechi.com IN NS ns2.linuxtechi.com
ns2.linuxtechi.com IN A 192.168.1.51
```

## NTP (Network Time Protocol)

The NTP system is based on Internet *time servers*, computers with access to atomic clocks such as those operated by the U.S. government. These NTP servers run a software service that provides the clock's time of day to client computers over UDP port 123. NTP supports a hierarchy of multiple server levels to handle a large load of client requests. The protocol includes algorithms to accurately adjust the time of day being reported to account for Internet network transmission delays.

Computers running Windows, Mac OS X and Linux operating systems can be configured to use an NTP server. Starting with Windows XP, for example, the Control Panel "Date and Time" option contains an Internet Time tab which allows choosing an NTP server and turning time synchronization on or off

The process name of the service	=	ntpd
Name of the rpm	=	ntpd

- First check to see if your OS has already service rpm package installed on the system

```
rpm -qa | grep ntp
ntp.x.x.x.x86_64
ntpdate.x.x.x86_64
```

- If the above rpms exist then you can proceed with the configuration
- If the rpms do not exist then download the package/rpm using yum command

```
yum install ntpd -y
```

- Once the package is installed make changes to the following configuration file:  
*/etc/ntp.conf* (Linux)

Add NTP server IP or hostname where server parameters are defined  
*server ntp-hostname*

- Now start the service

```
systemctl restart ntpd
```

- Also in order to have the service startup at the boot time run the following command

```
systemctl enable ntpd
```

- NTP Logs location

```
/var/log/messages
```

- Commands

```
ps -ef | grep ntp
```

OR

```
ntpq → peers → quit
```

```
ntpdate = Set date and time via ntp
```

```
ntpdate -v hostname = To manually force the time to sync with ntp server
```

- List of NTP Servers on Web:

<http://www.pool.ntp.org/zone/north-america>

## **rpm commands**

Syntax	Description	Example(s)
rpm -ivh {rpm-file}	Install a package	rpm -ivh mozilla-mail-1.7.5-17.i586.rpm rpm -ivh --test mozilla-mail-17.i586.rpm
rpm -Uvh {rpm-file}	Upgrade package	rpm -Uvh mozilla-mail-1.7.6-12.i586.rpm rpm -Uvh --test mozilla-mail-12.i586.rpm
rpm -ev {package}	Remove a package	rpm -ev mozilla-mail
rpm -ev --nodeps {package}	Remove package without che for dependencies	rpm -ev --nodeps mozilla-mail
rpm -qa	List all installed package	rpm -qa
rpm -qi {package}	Show installed package information along with description	rpm -qi mozilla-mail
rpm -qf {/pathToFile}	Find out what package ain/ File belongs to	rpm -qf /etc/passwd rpm -qf /bin/bash
rpm -qc {packagename}	List configuration file of a package	rpm -qc httpd
rpm -qcf {pathToFile}	List configuration files for a command	rpm -qcf /usr/X11R6/bin/xeyes
rpm -qa --list	List recently installed rpms	rpm -qa --list
Rjpm -qpR {.rpm-file}	List dependencies of a rpm	rpm - qpR Mozilla-mail rpm -qR bash

```
rpm -qa | grep tiger
```

```
yum install tigervnc*
```

```
yum remove tigervnc*
```

Go online and search for tigervnc  
Download the tigervnc rpm with wget command  
wget <http://centos.org/tiger.rpm>

```
rpm -hiv package.rpm
```

```
yum install tigervnc*
```

```
rpm -qa | grep tiger
```

```
rpm -qi | grep tiger.rpm
```

```
rpm -e tigervnc-license-1.8.0-5.el7.noarch
```

```
rpm -e tigervnc-server-1.8.0-5.el7.x86_64
```

```
rpm -e tiger-all.rpms
```

```
yum install tigervnc*
```

**List configuration files of a package**

```
rpm -qc tigervnc-server-1.8.0-5.el7.x86_64
```

**Find out which package this file/script belongs to**

```
rpm -qf /etc/sysconfig/vncservers
```

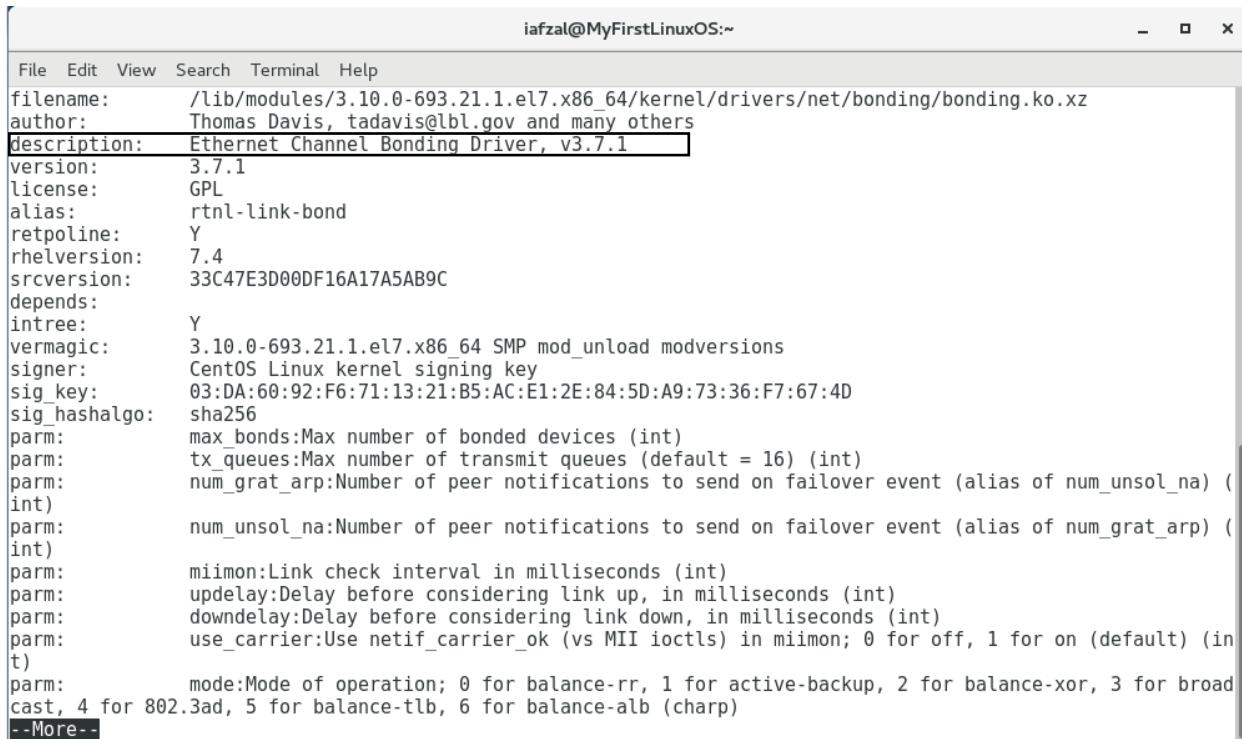
**Find out about pwd command**

```
rpm -qf /usr/bin/pwd
```

## Creating NIC Bonding:

- Add a new NIC if it does not exist
- Install bonding driver = **modprobe bonding**
- To list the bonding module info = **modinfo bonding**

*You will see the driver version as seen below if the driver is installed and loaded*



The screenshot shows a terminal window titled "iafzal@MyFirstLinuxOS:~". The window displays the output of the command "modinfo bonding". The output provides detailed information about the bonding kernel module, including its filename, author, description, version, license, alias, and various parameters and their descriptions.

```
iafzal@MyFirstLinuxOS:~
File Edit View Search Terminal Help
filename: /lib/modules/3.10.0-693.21.1.el7.x86_64/kernel/drivers/net/bonding/bonding.ko.xz
author: Thomas Davis, tadv...@lbl.gov and many others
description: Ethernet Channel Bonding Driver, v3.7.1
version: 3.7.1
license: GPL
alias: rtnl-link-bond
retpoline: Y
rhelversion: 7.4
srcversion: 33C47E3D00DF16A17A5AB9C
depends:
intree: Y
vermagic: 3.10.0-693.21.1.el7.x86_64 SMP mod_unload modversions
signer: CentOS Linux kernel signing key
sig_key: 03:DA:60:92:F6:71:13:21:B5:AC:E1:2E:84:5D:A9:73:36:F7:67:4D
sig_hashalgo: sha256
parm: max_bonds:Max number of bonded devices (int)
parm: tx_queues:Max number of transmit queues (default = 16) (int)
parm: num_grat_arp:Number of peer notifications to send on failover event (alias of num_unsol_na) (int)
parm: num_unsol_na:Number of peer notifications to send on failover event (alias of num_grat_arp) (int)
parm: miimon:Link check interval in milliseconds (int)
parm: updelay:Delay before considering link up, in milliseconds (int)
parm: downdelay:Delay before considering link down, in milliseconds (int)
parm: use_carrier:Use netif_carrier_ok (vs MII ioctl) in miimon; 0 for off, 1 for on (default) (int)
parm: mode:Mode of operation; 0 for balance-rr, 1 for active-backup, 2 for balance-xor, 3 for broadcast, 4 for 802.3ad, 5 for balance-tlb, 6 for balance-alb (charp)
--More--
```

## Create Bond Interface File

- vi /etc/sysconfig/network-scripts/ifcfg-bond0
- Add the following parameters

```
DEVICE=bond0
TYPE=Bond
NAME=bond0
BONDING_MASTER=yes
BOOTPROTO=none
ONBOOT=yes
IPADDR=192.168.1.80
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
BONDING_OPTS="mode=5 miimon=100"
```

- Save and exit the file
- The bonding options details are can be found on the following table

Mode	Policy	How it works	Fault Tolerance	Load balancing
0	Round Robin	packets are sequentially transmitted/received through each interfaces one by one.	No	Yes
1	Active Backup	one NIC active while another NIC is asleep. If the active NIC goes down, another NIC becomes active. only supported in x86 environments.	Yes	No
2	XOR [exclusive OR]	In this mode the, the MAC address of the slave NIC is matched up against the incoming request's MAC and once this connection is established same NIC is used to transmit/receive for the destination MAC.	Yes	Yes
3	Broadcast	All transmissions are sent on all slaves	Yes	No
4	Dynamic Link Aggregation	aggregated NICs act as one NIC which results in a higher throughput, but also provides failover in the case that a NIC fails. Dynamic Link Aggregation requires a switch that supports IEEE 802.3ad.	Yes	Yes
5	Transmit Load Balancing (TLB)	The outgoing traffic is distributed depending on the current load on each slave interface. Incoming traffic is received by the current slave. If the receiving slave fails, another slave takes over the MAC address of the failed slave.	Yes	Yes
6	Adaptive Load Balancing (ALB)	Unlike Dynamic Link Aggregation, Adaptive Load Balancing does not require any particular switch configuration. Adaptive Load Balancing is only supported in x86 environments. The receiving packets are load balanced through ARP negotiation.	Yes	Yes

### miimon

Specifies the MII link monitoring frequency in milliseconds. This determines how often the link state of each slave is inspected for link failures

Edit the First NIC File (enp0s3)

- vi /etc/sysconfig/network-scripts/ifcfg-enp0s3
- Delete the entire content
- Add the following parameters
 

```
TYPE=Ethernet
BOOTPROTO=none
DEVICE=enp0s3
ONBOOT=yes
HWADDR="MAC from the ifconfig command"
MASTER=bond0
SLAVE=yes
```
- Save and exit the file

Create the Second NIC File (enp0s8) or Copy enp0s3

- vi /etc/sysconfig/network-scripts/ifcfg-enp0s8
- Add the following parameters  
`TYPE=Ethernet  
BOOTPROTO=none  
DEVICE=enp0s8  
ONBOOT=yes  
HWADDR="MAC from the ifconfig command"  
MASTER=bond0  
SLAVE=yes`
- Save and exit the file

Restart the Network Service

- `systemctl restart network`

Test and verify the configuration

- `ifconfig`    or    `ifconfig | more`

Use following command to view bond interface settings like bonding mode & slave interface

- `cat /proc/net/bonding/bond0`

# YUM COMMAND CHEAT SHEET

for Red Hat Enterprise Linux

## YUM QUERIES

### SUBCOMMAND DESCRIPTIONS AND TASKS

**help** Display yum commands and options  
**yum help** Show yum subcommands and options

### Individual packages

**list** List package names from repositories  
**yum list available** List all available packages  
**yum list installed** List all installed packages  
**yum list all** List installed and available packages  
**yum list kernel** List installed and available kernel packages

**info** Display information about a package  
**yum info vsftpd** List info about vsftpd package

**deplist** Display dependencies for a package  
**yum deplist nfs-utils** List dependencies and packages providing them

**provides** Find packages that provide the queried file  
**yum provides “\*bin/top”** Show package that contains top command  
**yum provides “\*/README.top”** Show package containing README.top file

**search** Search package names and descriptions for a term  
**yum search samba** Find packages with samba in name or description

**updateinfo** Get information about available package updates  
**yum updateinfo security** Get info on available security updates

### Groups of packages

**grouplist** List names of installed and available package groups  
**groupinfo** Display description and contents of a package group  
**yum groupinfo “Web Server”** See packages in Web Server group

**check-update** Query repositories for available package updates

## MANAGE YUM REPOSITORIES

### SUBCOMMAND DESCRIPTIONS AND TASKS

**repolist** Display enabled software repositories

**repoinfo** Display information about enabled yum repositories \*  
**yum repoinfo rhel-7-server-rpms** See info on rhel-7-server-rpms repo

**repo-pkgs** Work with packages in a particular repository \*  
**yum repo-pkgs my-rpms list** List packages from my-rpms repo  
**yum repo-pkgs my-rpms install** Install all packages from my-rpms repo  
**yum repo-pkgs my-rpms remove** Remove all packages from my-rpms repo

**makecache** Download yum repository data to cache

## TROUBLESHOOT AND MAINTAIN YUM

### SUBCOMMAND DESCRIPTIONS AND TASKS

**check** Check the local RPM database for problems (runs for a long time)

**history** View and use yum transactions  
**yum history list** List all yum install, update and erase actions

**yum history info 3** Show details of yum transaction 3

**yum history undo 3** Undo the yum action from transaction 3

**yum history redo 3** Redo the undone yum action from transaction 3

**clean** Clear out cached package data

**yum clean packages** Delete packages saved in cache

**yum clean all** Clean out all packages and meta data from cache

**fssnapshot** List LVM snapshots (helps roll back after package updates)

**fs** Act on filesystem (prevent doc or language file install on minimal systems)

**yum fs filters** List enabled filesystem filters

**yum fs documentation** Filters all docs from being installed (careful!)

## INSTALL, REMOVE AND UPGRADE PACKAGES WITH YUM

### SUBCOMMAND DESCRIPTIONS AND TASKS

**install** Install a package from a repository to your system  
**yum install vsftpd** Install the vsftpd package

**update** Update one or all packages on your system  
**yum update** Update all packages with available updates  
**yum update httpd** Update the httpd package (if available)  
**yum update --security** Apply security-related package updates

**update-to** Update one or all packages to a particular version

**upgrade** Update packages taking obsoletes into account

**localinstall** Install a package from a local file, http, or ftp  
**yum localinstall abc-1-1.i686.rpm** Install abc package from local directory  
**yum localinstall http://myrepo/abc-1-1.i686.rpm** Install abc from FTP site

**downgrade** Downgrade a package to an earlier version  
**yum downgrade abc** Downgrade the abc package to an earlier version

**reinstall** Reinstall the current version of a package  
**yum reinstall util-linux** Reinstall util-linux (to replace any deleted files)

**swap** Remove one package and install another  
**yum swap ftp lftp** Remove ftp package and install lftp package

**erase** Erase a package (and possibly dependencies) from your system  
**yum remove vsftpd** Remove the vsftpd package and dependencies

**remove** Same as erase

**autoremove** Same as erase, plus removes additional unneeded packages \*  
**yum autoremove httpd** Remove httpd and other unneeded packages

**groupinstall** Install all packages in the selected group  
**yum groupinstall “Web server”** Install Web Server packages

## MANAGE LANGUAGE PACKAGES WITH YUM

SUBCOMMAND	DESCRIPTIONS AND TASKS
<b>langavailable</b>	List all available languages *
<b>langinfo</b>	List packages available for a language * <code>yum langinfo es</code> List packages associated with Spanish language
<b>langinstall</b>	Install packages associated with a particular language * <code>yum langinstall es</code> Install packages associated with Spanish language
<b>langlist</b>	List languages that are installed *
<b>langremove</b>	Remove installed language packs for a language * <code>yum langremove es</code> Remove packages associated with Spanish language

## POPULAR OPTIONS FOR DIFFERENT YUM COMMANDS \*\*

OPTION	DESCRIPTION
<b>-y</b>	Assume yes if prompted
<b>--assumeno</b>	Assume no if prompted
<b>-q</b>	Produce no output
<b>-v</b>	Produce extra debugging output
<b>--nopugins</b>	Run command without loading any yum plugins
<b>--disableplugin=</b>	Disable a particular plugin for single command <code>yum --disableplugin=langpacks info vsftpd</code>
<b>--enableplugin=</b>	Enable a plugin that is installed, but currently disabled <code>yum --enableplugin=ps ps</code> Show packages tied to running processes
<b>--enablerepo=</b>	Enable currently disabled repo for a single command (wildcards okay) <code>yum install docker --enablerepo=rhel-7-server-extras-rpm</code>
<b>--disablerepo=</b>	Disable currently enabled repo for a single command (wildcards okay) <code>yum list available --disablerepo=epel</code>
<b>--downloadonly</b>	Download to /var/cache/yum/arch/prod/repo/ packages/, but don't install <code>yum install --downloadonly vsftpd</code> Download vsftpd package to cache
<b>--filter-???</b>	Replace ??? with vendors, rpm-groups, arches, and others to filter output
<b>--changelog</b>	Display changelog information of package

## MORE YUM-RELATED COMMANDS (install the yum-utils package)

COMMAND	DESCRIPTION
<b>find-repos-of-install</b>	Find which repository a package comes from
<b>needs-restarting</b>	Find processes that have been updated and need to restart
<b>repoclosure</b>	Get unmet dependency list from repositories
<b>repoquery</b>	Query remote repos and local RPM database <code>repoquery --requires --resolve bash</code> Show dependent packages
<b>reposync</b>	Synchronize yum repositories to a local directory <code>reposync -r rhel-atomic-host-beta-rpms</code> Get packages from repo
<b>repotrack</b>	Download a package and all its dependencies
<b>show-installed</b>	List installed RPM packages and statistics
<b>verifytree</b>	Check the local yum repository for consistency
<b>yum-complete-transaction</b>	Try to complete yum transactions that didn't finish
<b>yumdb</b>	Check or change the yum database
<b>yumdownloader</b>	Download a package from a repo to current directory

Type `man yum` for further details on all yum subcommands and options

\* New options for RHEL 7

\*\* Some options need yum plugins. Type `yum list "yum-plugin*"` to see available plugins.

- User Account
  - User account naming convention
  - User account user IDs
  - User password policies –
 

```
chage -l username
chage -help
/etc/shadow
vi /etc/login.defs
```
  - Disable old password
 

```
cd /etc/pam.d/system-auth
```
  - User or service account files and directories permission
- Remove un-wanted packages
  - Install what you need
  - Remove packages no longer in use
- Stop un-used Services
  - List all running services
 

```
systemctl (List only running service)
systemctl -a (List every service running or not)
```
  - telnet, ftp, NFS etc.
- Check on Listening Ports
  - netstat -tunlp
- Secure SSH Configuration
  - Disable direct root login
  - Change SSH port
- Enable Firewall (iptables/firewalld)
  - Older version = iptables
  - New version = firewalld
 

```
firewall-config (GUI)
firewall-cmd
iptables
```
  - older version = /etc/sysconfig/iptables-config
  - new version = /etc/firewalld/
- Enable SELinux

Security-Enhanced Linux (SELinux) is a security architecture integrated into the 2.6.x kernel using the Linux Security Modules (LSM). It is a project of the United States National Security Agency (NSA) and the SELinux community. SELinux integration into Red Hat Enterprise Linux was a joint effort between the NSA and Red Hat.

SELinux defines the access and transition rights of every user, application, process, and file on the system

/etc/sysconfig/selinux

enforcing — The SELinux security policy is enforced.

permissive — The SELinux system prints warnings but does not enforce policy.

This is useful for debugging and troubleshooting purposes.

disabled — SELinux is fully disabled. SELinux hooks are disengaged from the kernel and the pseudo-file system is unregistered.

Commands = sestatus

Find status of a file = stat filename

Other commands = chcon, checkpolicy, newrole, getsebool, setsebool, fixfiles, semanage

Documentation attached within the hand-out section

- Change Listening Services Port Numbers
- Keep your OS up to date (patching)

## Create local repository

First, mount your CentOS 7 installation DVD. For example, let us mount the installation media on /mnt directory.

- `mount /dev/cdrom /mnt/`
- `cd /mnt/Packages/`

We need a package called “createrepo” to create our local repository. So let us install it too.

If you did a minimal CentOS installation, then you might need to install the following dependencies first.

- `rpm -ivh libxml2-python-2.9.1-5.el7.x86_64.rpm`
- `rpm -ivh deltarpm-3.6-3.el7.x86_64.rpm`
- `rpm -ivh python-deltarpm-3.6-3.el7.x86_64.rpm`

Now install “createrepo” package:

- `rpm -ivh createrepo-0.9.9-23.el7.noarch.rpm`

### Build Local Repository

It's time to build our local repository. Create a storage directory to store all packages from CentOS DVD's.

- `mkdir /tmp/localrepo`

Now, copy all the files from CentOS DVD(s)

- `cp -ar /mnt/Packages/* /tmp/localrepo`

Once you copied all the files, create a repository file called “localrepo.repo” under /etc/yum.repos.d/ directory and add the following lines into the file. You can name this file as per your liking.

- `vi /etc/yum.repos.d/localrepo.repo`

Add the following lines.

```
[localrepo]
name=Unixmen Repository
baseurl=file:///tmp/localrepo
```

```
gpgcheck=0
enabled=1
```

Note: Use three slashes(///) in the baseurl.

Now, start building local repository:

- createrepo -v /tmp/localrepo/

Now the repository building process will start.

Now, list out the repositories using the following command.

- yum repolist

Clean the Yum cache and update the repository lists.

- yum clean all
- yum update

After creating the repository, disable or rename the existing repositories if you only want to install packages from the local repository itself.

- yum install --disablerepo="\*" --enablerepo="localrepo" httpd

## How to Create Local YUM Repository on CentOS / RHEL Operating System

### Step 1:

Mount the local media (CD or DVD)

```
mount /dev/cdrom /mnt
```

### Step-2:

Copy or Extract the Media

```
Become root
cd /
mkdir localrepo
cp -rv /media/* /localrepo/
```

### Step-3:

Remove the Online Repository

```
rm -rf /etc/yum.repos.d/*
```

### Step-4:

Create Local Repository File

```
vi /etc/yum.repos.d/local.repo
```

Add (For CentOS)

```
[centos7]
name=centos7
baseurl=file:///localrepo/
enabled=1
gpgcheck=0
```

Add (For Redhat)

```
[rhel7]
name=rhel7
baseurl=file:///localrepo/
enabled=1
gpgcheck=0
```

### Step-5:

Update the local Repository database

```
createrepo /localrepo/
```

**Step 6.**

Clean repository cache

```
yum clean all
```

**Step 7.**

Verify newly created repository

```
yum repolist all
```

**Step 8.**

Test Repository

```
yum install tomcat
```

## SSH without a Password

### On Server

Run

**ssh-keygen** = Enter to all prompts

*This will generate a long SSH key*

### On Client

Run

**mkdir /root/.ssh**      *(If .ssh does not exist)*

Add the server keys to

**/root/.ssh/authorized\_keys**

## SCP examples.

- Copy file from a remote host to local host SCP example:

```
$ scp username@from_host:file.txt /local/directory/
```

- Copy file from local host to a remote host SCP example:

```
$ scp file.txt username@to_host:/remote/directory/
```

- Copy directory from a remote host to local host SCP example:

```
$ scp -r username@from_host:/remote/directory/ /local/directory/
```

- Copy directory from local host to a remote host SCP example:

```
$ scp -r /local/directory/ username@to_host:/remote/directory/
```

- Copy file from remote host to remote host SCP example:

```
$ scp username@from_host:/remote/directory/file.txt
username@to_host:/remote/directory/
```

# **HANDOUTS**

# **MODULE 8**

# **WELCOME TO: MODULE 8**

**DISK MANAGEMENT AND  
RUN LEVELS**

# System Run Level

- System Run Levels

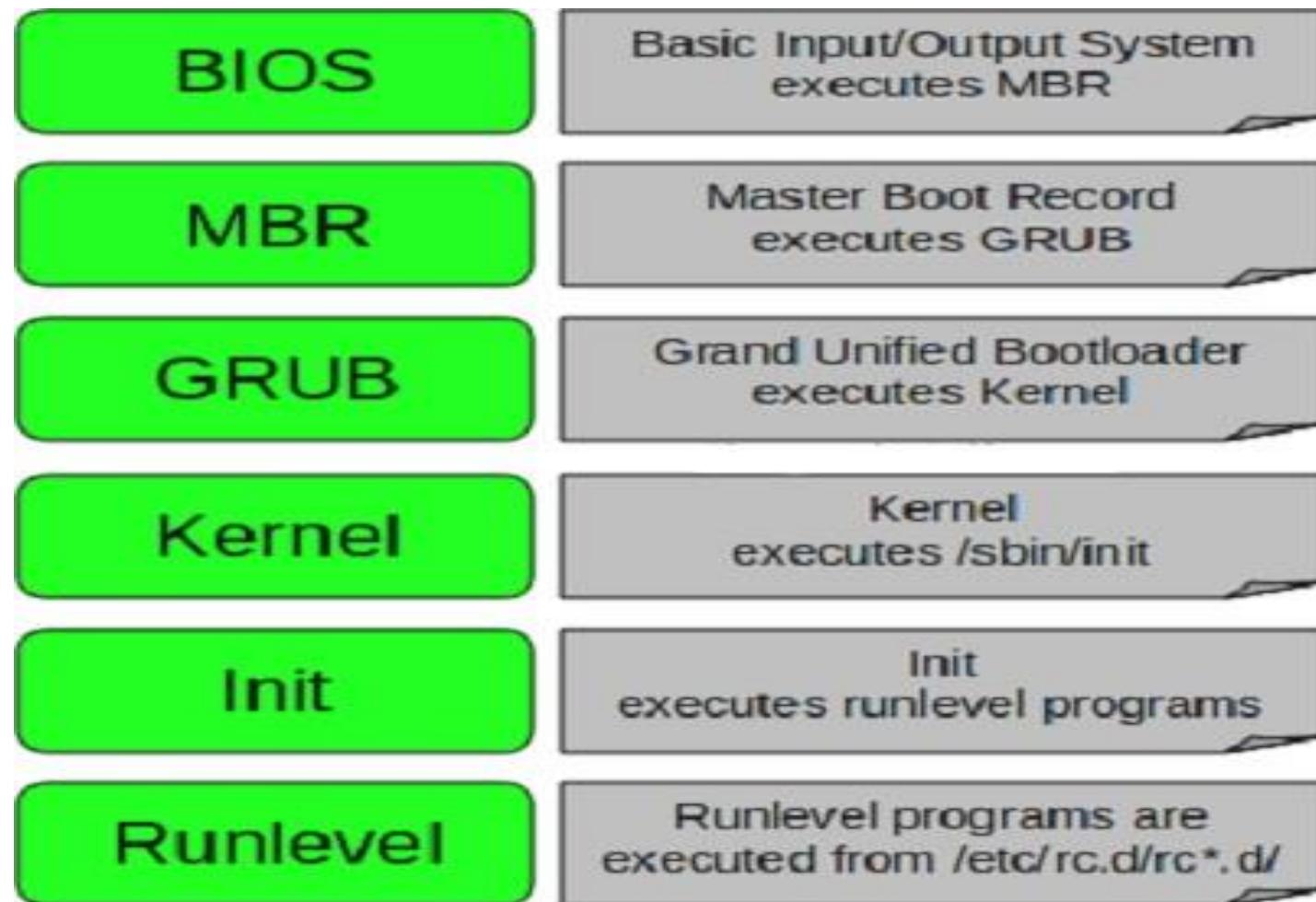
## *Main Run level*

- 0 Shut down (or halt) the system
- 1 Single-user mode; usually aliased as s or S
- 6 Reboot the system

## *Other Run levels*

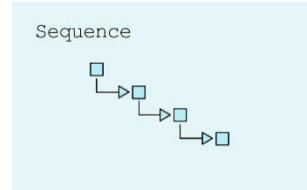
- 2 Multiuser mode without networking
- 3 Multiuser mode with networking
- 5 Multiuser mode with networking and GUI.

# Linux Boot Process



# Linux Boot Process (Newer Versions)

- The boot sequence changes in CentOS/Redhat 7 and above
- **systemd** is the new service manager in CentOS/RHEL 7 that manages the boot sequence
- It is backward compatible with SysV init scripts used by previous versions of RedHat Linux including RHEL 6
- Every system administrator needs to understand the boot process of an OS in order to troubleshoot effectively



# Linux Boot Process (Newer Versions)

BIOS = Basic Input and Output Setting (firmware interface)

POST = Power-On Self-Test started

**MBR = Master Boot Record**

Information saved in the first sector of a hard disk that indicates where the GRUB2 is located so it can be loaded in computer RAM

**GRUB2 = Grand Unified Boot Loader v2**  
Loads Linux kernel  
`/boot/grub2/grub.cfg`

**Kernel = Core of Operating System**  
Loads required drivers from `initrd.img`  
Starts the first OS process (`systemd`)

**Systemd = System Daemon (PID # 1)**  
It then starts all the required processes  
Reads = `/etc/systemd/system/default.target` to bring the system to the run-level  
Total of 7 run-levels (0 thru 6)

# Message of the Day

- Message of the day file location
  - **/etc/motd**

# Customize Message of the Day

- Once again, message of the day is the first message users will see when they login to the Linux machine
- Steps:
  - Create a new file in `/etc/profile.d/motd.sh`
  - Add desired commands in motd.sh file
  - Modify the `/etc/ssh/sshd_config` file to edit  
`#PrintMotd yes` to `PrintMotd no`
  - Restart sshd service
    - `systemctl restart sshd.service`



# Disk Partition

- Commands for disk partition
  - **df**
  - **fdisk**

# **Adding Disk and Creating Partition**

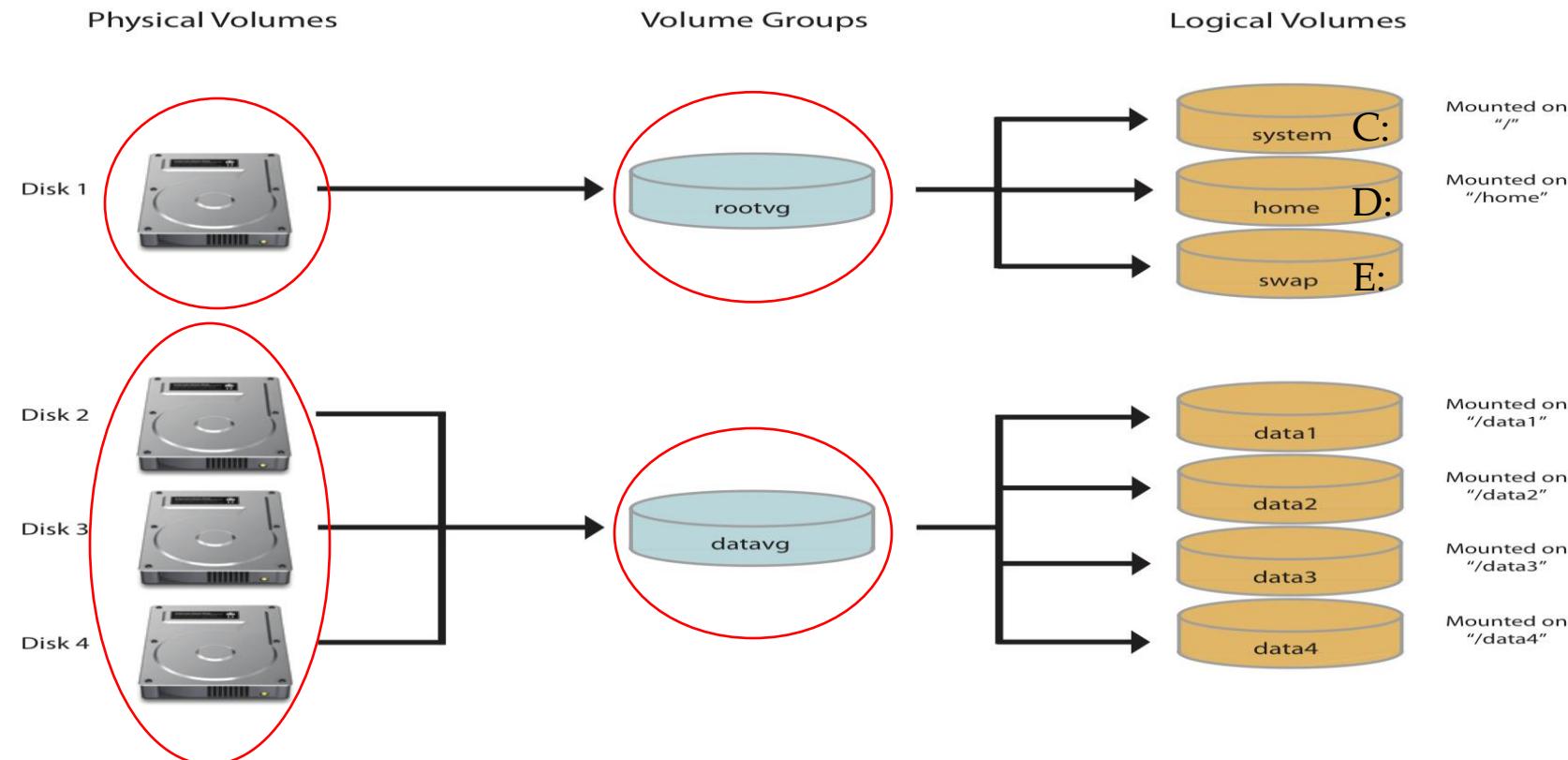
- Purpose? = Out of Space, Additional Apps etc.
- Commands for disk partition
  - **df**
  - **fdisk**

# Computer Storage

- Local Storage
  - RAM, HDD, SSD, etc.
- DAS (Direct Attached Storage)
  - CD/DVD, USB flash drive, external disk directly attached with USB or other cables
- SAN (Storage Area Network)
  - Storage attached through iSCSI or fiber cable
- NAS (Network Attached Storage)
  - Storage attached over network (TCP/IP)
  - E.g. Samba, NFS etc.

# Logical Volume Management (LVM)

- LVM allows disks to be combined together



# LVM Configuration During Install

- Install Linux CentOS with LVM configuration



# Add Disk and Create LVM Partition

File System

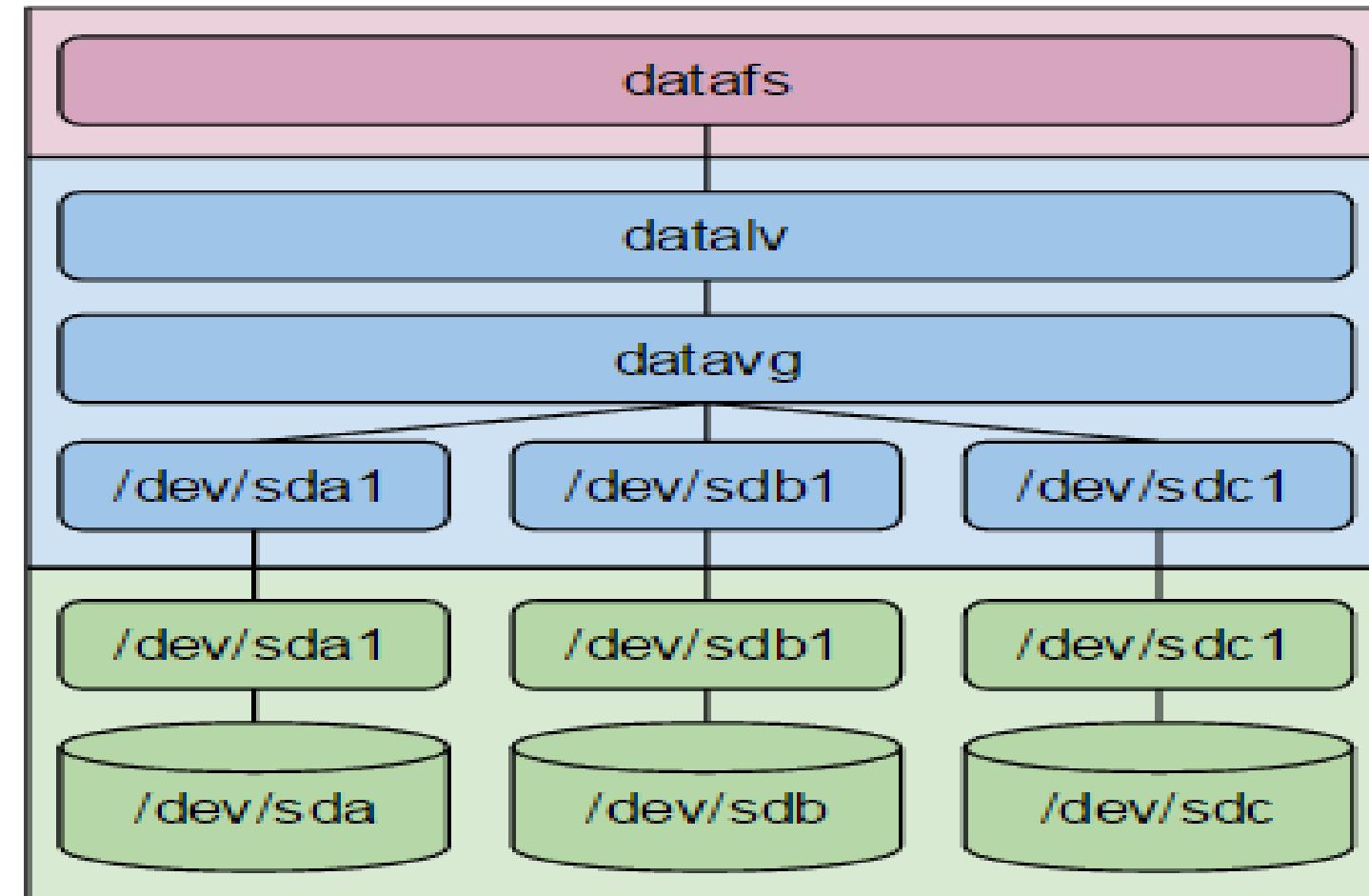
Logical Volume(s)

Volume Group

Physical Volume

Partitions

Hard Disks



# ADD AND EXTEND DISK USING LVM

/oracle = 1.0G

/oracle = Full

Few Options:

- Delete older files to free up disk space
- Add new physical disk mount to /oracle2
- Create a new virtual disk and mount to /oracle2
- Or extend /oracle through LVM.

# ADD/EXTEND SWAP SPACE

- **What is swap? – CentOS.org**

Swap space in Linux is used when the amount of physical memory (RAM) is full. If the system needs more memory resources and the RAM is full, inactive pages in memory are moved to the swap space. While swap space can help machines with a small amount of RAM, it should not be considered a replacement for more RAM. Swap space is located on hard drives, which have a slower access time than physical memory

- **Recommended swap size = Twice the size of RAM**

M = Amount of RAM in GB, and S = Amount of swap in GB, then

If  $M < 2$

then  $S = M * 2$

Else  $S = M + 2$

- **Commands**

- dd
- mkswap
- swapon or swapoff

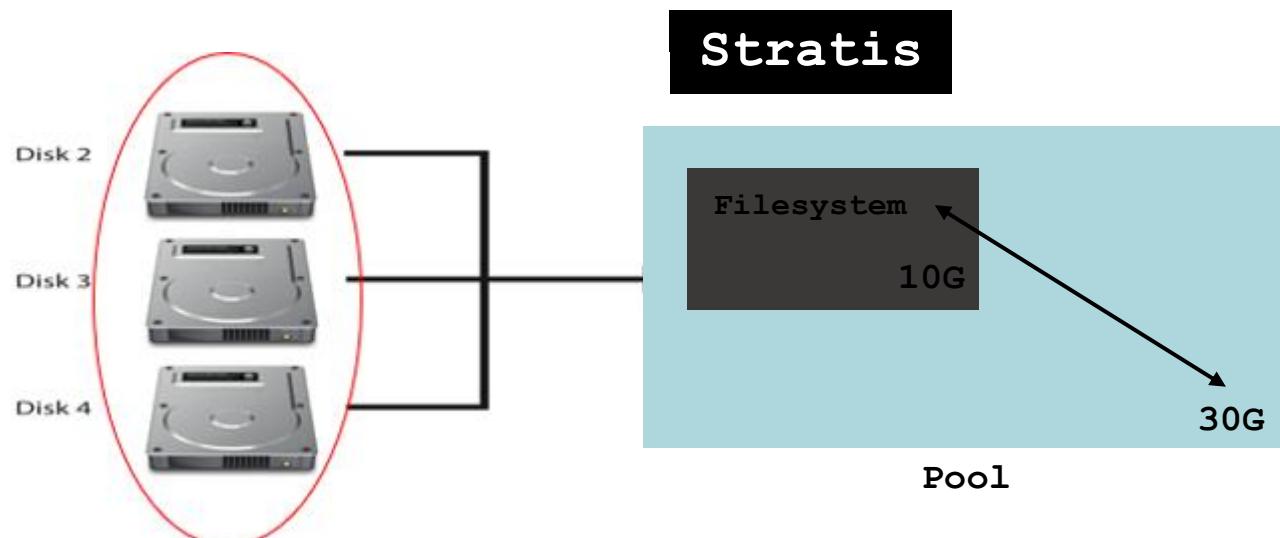
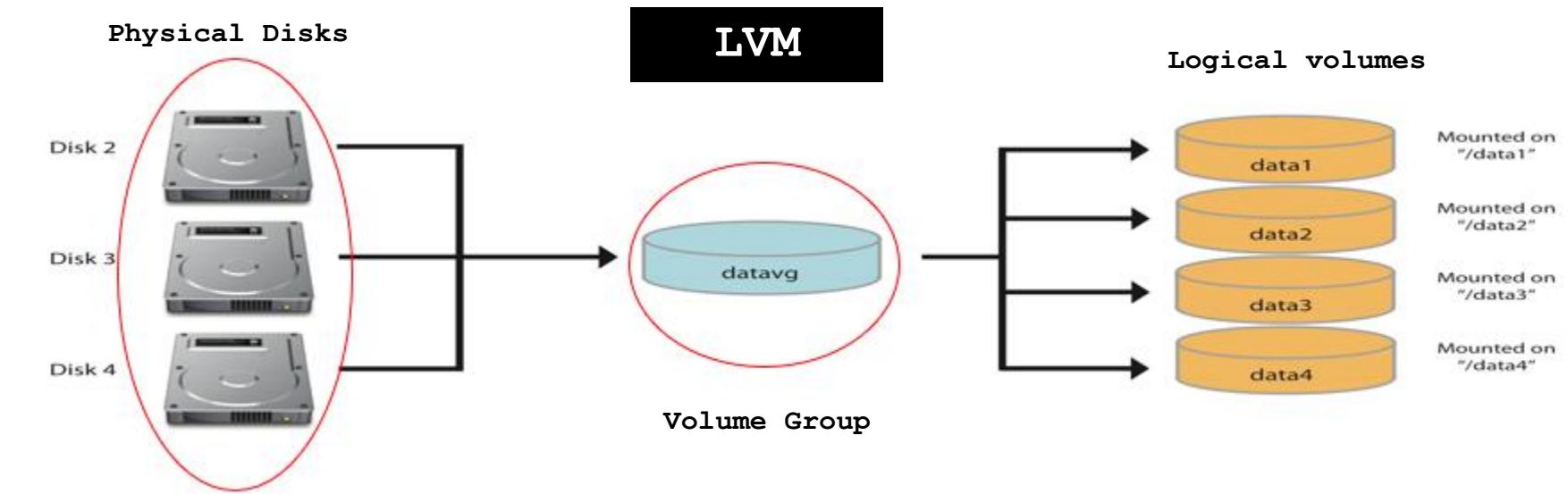
# Implement Advanced Storage Features

- Red Hat 8 introduces the next generation volume management solution called Stratis
- It uses thin provisioning by default
- It combines the process of creating logical volume management (LVM) and creation of filesystems into one management
- In LVM if a filesystem system gets full you will have to extend it manually whereas stratis extends the filesystem automatically if it has available space in its pool

In this lesson we will learn...

- How to manage multiple storage layers using Stratis local storage management

# Implement Advanced Storage Features



# Implement Advanced Storage Features

- Install Stratis package

```
yum/dnf install stratis-cli stratisd
```

- Enable and start Stratis service

```
systemctl enable|start stratisd
```

- Add 2 x 5G new disks from virtualization software and verify at the OS level

```
Oracle virtualbox storage setting
lsblk
```

- Create a new stratis pool and verify

```
stratis pool create pool1 /dev/sdb
stratis pool list
```

- Extend the pool

```
stratis pool add-data pool1 /dev/sdc
stratis pool list
```

# Implement Advanced Storage Features

- Create a new filesystem using stratis

```
stratis filesystem create pool1 fs1
stratis filesystem list (Filesystem will start with 546 MB)
```

- Create a directory for mount point and mount filesystem

```
mkdir /bigdata
mount /dev/stratis/pool1/fs1 /bigdata
lsblk
```

- Create a snapshot of your filesystem

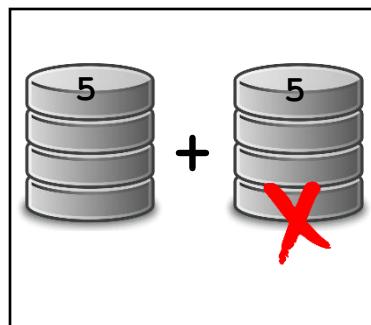
```
stratis filesystem snapshot pool1 fs1 fs1-snap
stratis filesystem list
```

- Add the entry to /etc/fstab to mount at boot

```
UUID="af8-0887afgdja-" /fs1 xfs defaults,x-
systemd.requires=stratisd.service 0 0
```

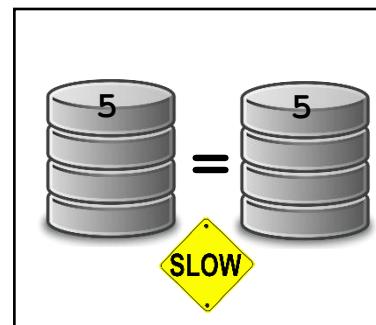
# RAID

- RAID (Redundant Array of Independent Disks)
- Type of RAID
  - RAID0
  - RAID1
  - RAID5



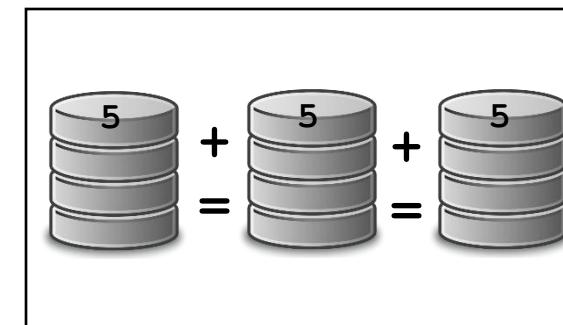
RAID0

$$5+5=10G$$



RAID1

$$5,5=5G$$



RAID5

$$5+5+5=12G$$

# File System Check (fsck and xfs\_repair)



- Linux fsck utility is used to check and repair Linux filesystems (ext2, ext3, ext4, etc.)
- Linux xfs\_repair utility is used to check and repair Linux filesystems for xfs filesystem type
- Depending on when was the last time a file system was checked, the system runs the fsck during boot time to check whether the filesystem is in consistent state
- System administrator could also run it manually when there is a problem with the filesystems
- Make sure to execute the fsck on an **unmounted** file systems to avoid any data corruption issues.

# File System Check (fsck and xfs\_repair)...



- Force a filesystem check even if it's clean using option `-f`
- Attempt to fix detected problems automatically using option `-y`
- The `xfs_repair` utility is highly scalable and is designed to repair even very large file systems with many inodes efficiently. Unlike other Linux file systems, `xfs_repair` does not run at boot time
- The following are the possible **exit codes** for `fsck` command
  - 0 – No errors
  - 1 – Filesystem errors corrected
  - 2 – System should be rebooted
  - 4 – Filesystem errors left uncorrected
  - 8 – Operational error
  - 16 – Usage or syntax error
  - 32 – Fsck canceled by user request
  - 128 – Shared-library error

# System Backup (dd Command)

## 5 Different Types of Backups

1. System backup (entire image using tools such as acronis, Veeam, Commvault etc.)
  2. Application backup (3<sup>rd</sup> party application backup solution)
  3. Database backup (Oracle dataguard, SQL backup etc.)
  4. Filesystem backup (tar, gzip directoris etc.)
  5. Disk backup or disk cloning (dd command)
- 
- dd is a command-line utility for Unix and Unix-like operating systems whose primary purpose is to convert and copy files
  - As a result, dd can be used for tasks such as backing up the boot sector of a hard drive, and obtaining a fixed amount of random data
  - Please note the source and destination disk should be the same size

# System Backup (dd Command)...

- To backup or clone an entire hard disk to another hard disk connected to the same system, execute the dd command as shown

```
dd if=<source file name> of=<target file name> [Options]
dd if=/dev/sda of=/dev/sdb
```

- To backup/copy the disk partition

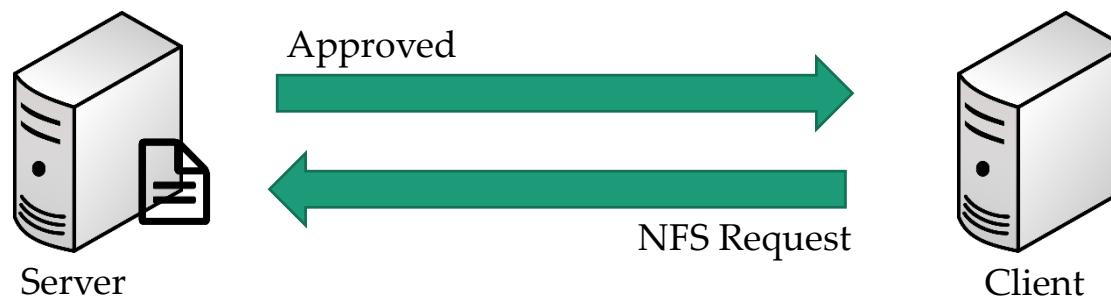
```
dd if=/dev/sda1 of=/root/sda1.img
```

- Restoring this image file to other machine after copying the .img

```
dd if=/root/sda1.img of=/dev/sdb3
```

# Network File System (NFS)

- NFS stands for Network File System, a file system developed by Sun Microsystems, Inc.
- It is a client/server system that allows users to access files across a network and treat them as if they resided in a local file directory
- For example, if you were using a computer linked to a second computer via NFS, you could access files on the second computer as if they resided in a directory on the first computer. This is accomplished through the processes of exporting (the process by which an NFS server provides remote clients with access to its files) and mounting (the process by which client map NFS shared filesystem)



# Network File System (NFS)...

## Steps for NFS Server Configuration

- Install NFS packages

```
yum install nfs-utils libnfsidmap (most likely they are installed)
```

- Once the packages are installed, enable and start NFS services

```
systemctl enable rpcbind
```

```
systemctl enable nfs-server
```

```
systemctl start rpcbind, nfs-server, rpc-statd, nfs-idmapd
```

- Create NFS share directory and assign permissions

```
mkdir /mypretzels
```

Read/write

all changes to the according filesystem are immediately flushed to disk; the respective write operations are being waited for

```
chmod a+rwx /mypretzels
```

- Modify **/etc/exports** file to add new shared filesystem

```
/mypretzels 192.168.12.7 (rw,sync,no_root_squash) = for only 1 host
```

```
/mypretzels *(rw,sync,no_root_squash) = for everyone
```

NFS share

IP address of  
client machine

# exportfs -rv

root on the client machine will have the same level of access to the files on the system as root on the server.

# Network File System (NFS)...

## Steps for NFS Client Configuration

- Install NFS packages

```
yum install nfs-utils rpcbind
```
- Once the packages are installed enable and start rpcbind service

```
systemctl rpcbind start
```
- Make sure firewalld or iptables stopped (if running)

```
ps -ef | egrep "firewall|iptable"
```
- Show mount from the NFS server

```
showmount -e 192.168.1.5 (NFS Server IP)
```
- Create a mount point

```
mkdir /mnt/kramer
```
- Mount the NFS filesystem

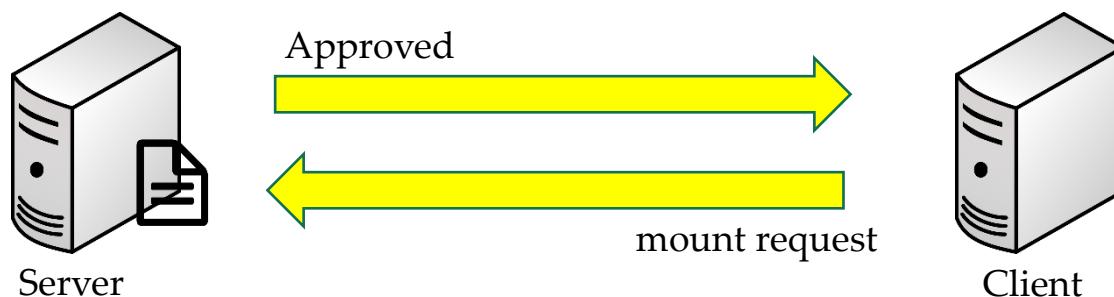
```
mount 192.168.1.5:/mypretzels /mnt/kramer
```
- Verify mounted filesystem

```
df -h
```
- To unmount

```
umount /mnt/kramer
```

# Samba

- Samba is a Linux tool or utility that allows sharing for Linux resources such as files and printers to with other operating systems
- It works exactly like NFS but the difference is NFS shares within Linux or Unix like system whereas Samba shares with other OS (e.g. Windows, MAC etc.)
- For example, computer “A” shares its filesystem with computer “B” using Samba then computer “B” will see that shared filesystem as if it is mounted as the local filesystem



# Samba (smb vs. CIFS)

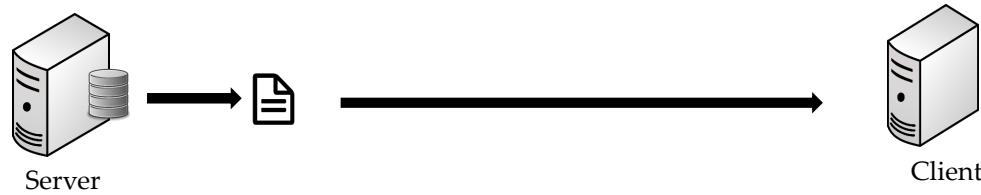
- Samba shares its filesystem through a protocol called **SMB** (Server Message Block) which was invented by IBM
- Another protocol used to share Samba is through **CIFS** (Common Internet File System) invented by Microsoft and NMB (NetBios Named Server)
- **CIFS** became the extension of **SMB** and now Microsoft has introduced newer version of SMB v2 and v3 that are mostly used in the industry
- In simple term, most people, when they use either SMB or CIFS, are talking about the same exact thing

# Samba Installation and Configuration

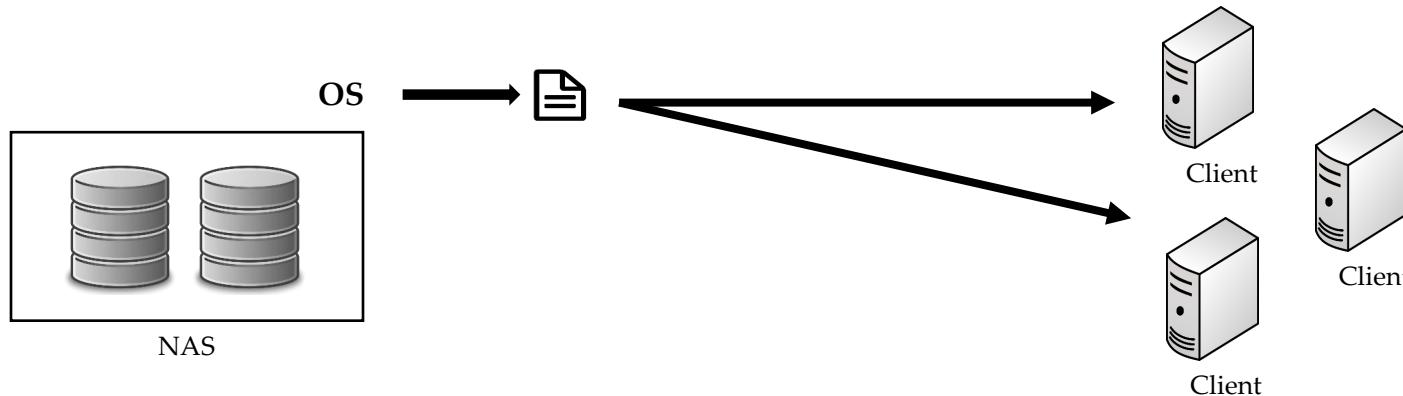
- Take snapshot of your VM
- Install samba packages
- Enable samba to be allowed through firewall (Only if you have firewall running)
- Disable firewall
- Create Samba share directory and assign permissions
- Also change the SELinux security context for the samba shared directory
- Or disable SELinux
- Modify `/etc/samba/smb.conf` file to add new shared filesystem
- Verify the setting
- Once the packages are installed, enable and start **Samba** services (smb and nmb)
- Mount Samba share on Windows client
- Mount Samba share on Linux client
- Additional instructions on creating secure Samba share.

# NAS Device for NFS or Samba

- A storage can be carved on a Linux server, and it can be shared with another Linux machine through NFS or to a Windows machine through Samba service



- NFS/Samba or any NAS service can be setup through a dedicated NAS device



# NAS Device for NFS or Samba

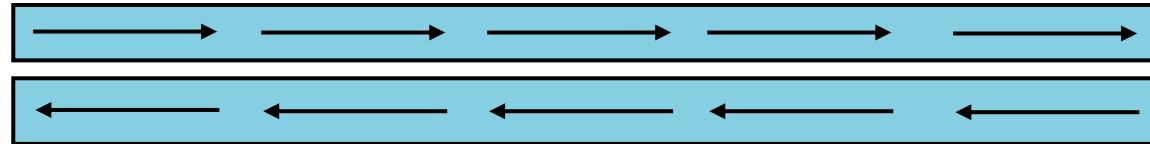
- In this video we will learn...
  - Physical layout of a NAS device
  - Setup, configure and manage NAS device
  - Create shared filesystem (NFS and Samba)
  - Mount shared folder from the NAS device to Linux and Windows

# SATA and SAS

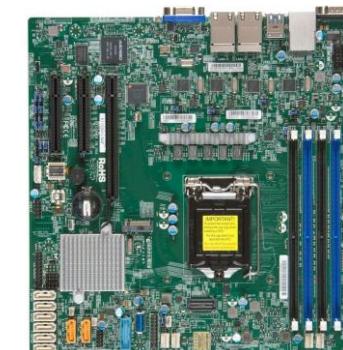
- **SATA** Stands for Serial Advanced Technology Attachment and **SAS** stands for Serial Attached SCSI (SCSI Stands for Small Computer System Interface, typically pronounced as “scuzzy”)
- Both SAS and SATA utilize serial communication. Serial communication means that the highway has both lanes



Parallel communication



Serial communication



# SATA and SAS

- The main difference between them is that SAS drives are faster and more reliable than SATA drives
- SAS is generally more expensive, and it's better suited for use in servers or in processing-heavy computer workstations. SATA is less expensive, and it's better suited for desktop file storage
- In a SATA cable, all 4 wires are placed within the same cable. In a SAS cable, the 4 wires are separated into 2 different cables

Why divide the wires between 2 cables???

- So you can connect more devices to one another. With a SATA cable, you can only link the motherboard and the storage drive. You could hook up an expansion device, but that takes up valuable room inside your computer.
- With a SAS cable, you can hook up the motherboard to both a storage drive and another piece of hardware that has SAS connectors.

**Here's what the highways look like, metaphorically:**

- SATA cable:** Los Angeles to San Francisco
- SAS cable:** Los Angeles to San Francisco or Los Angeles to Las Vegas

# Difference Between CentOS/RHEL 7 and 8

- Red Hat Enterprise Linux 8 (RHEL 8) is now available for production use with lots of developer-friendly capabilities
- RHEL 8 official release by Red Hat Inc, was announced on **May 7, 2019**
- I will cover only what is changed and what you should in terms of my Linux course

	RHEL 8	RHEL 7
General Availability Date	14-Nov-18	10-Jun-14
Code Name	Ootpa	Maipo
Kernel Version	4.18	3.10.0-123
End of Support	May-2029	30-Jun-2024
Last Minor Release	8.x	7.7
Network Time Synchronization	Only Chrony	Chrony and ntpd
GUI Interface (Desktop)	Gnome 3.28	Gnome 3
Default Database	MySQL 8.0, MariaDB 10.3, PostgreSQL 10 and 9.6, and Redis 5.0	MariaDB

# Difference Between CentOS/RHEL 7 and 8

	RHEL 8	RHEL 7
Default Firewall	Firewalld, it uses nftables framework in the backend	Firewalld, it uses Iptables framework in the backend
Max Supported (Individual) File & Filesystem Size	XFS= 1024TB	XFS= 500TB
Package Management	By default both are installed, YUM symbolic link to DNF	By default only YUM and DNF can be installed from the Extra repo
Max. RAM Supported	24 TB on x86_64 architecture	12 TB on x86_64 architecture

## **System Run Level**

A run level is a preset operating state on a Unix-like operating system.

A system can be booted into (i.e., started up into) any of several runlevels, each of which is represented by a single digit integer. Each runlevel designates a different system configuration and allows access to a different combination of processes (i.e., instances of executing programs).

There are differences in the runlevels according to the operating system. Seven runlevels are supported in the standard Linux kernel (i.e., core of the operating system). They are:

0 - System halt; no activity, the system can be safely powered down.

1 - Single user; rarely used.

2 - Multiple users, no NFS (network filesystem); also used rarely.

3 - Multiple users, command line (i.e., all-text mode) interface; the standard runlevel for most Linux-based server hardware.

4 - User-definable

5 - Multiple users, GUI (graphical user interface); the standard runlevel for most Linux-based desktop systems.

6 - Reboot; used when restarting the system.

By default Linux boots either to runlevel 3 or to runlevel 5. The former permits the system to run all services except for a GUI. The latter allows all services including a GUI.

In addition to the standard runlevels, users can modify the preset runlevels or even create new ones if desired. Runlevels 2 and 4 are usually used for user defined runlevels.

The program responsible for altering the runlevel is init, and it can be called using the telinit command. For example, changing from runlevel 3 to runlevel 5, which allows the GUI to be started, can be accomplished by the root (i.e., administrative) user by issuing the following command:

```
telinit 5
```

Booting into a different runlevel can help solve certain problems. For example, if a change made in the X Window System configuration on a machine that has been set up to boot into a GUI has rendered the system unusable, it is possible to temporarily boot into a console (i.e., all-text mode) runlevel (i.e., runlevels 3 or 1) in order to repair the error and then reboot into the GUI. The X Window System is a widely used system for managing GUIs on single computers and on networks of computers.

Likewise, if a machine will not boot due to a damaged configuration file or will not allow logging in because of a corrupted /etc/passwd file (which stores user names and other data about users) or because of a forgotten password, the problem can be solved by first booting into single-user mode (i.e. runlevel 1).

The runlevel command can be used to find both the current runlevel and the previous runlevel by merely typing the following and pressing the Enter key:

```
/sbin/runlevel
```

The runlevel executable file (i.e., the ready-to-run form of the program) is typically located in the /sbin directory, which contains mostly administrative tools and which by default is not in the user's PATH (i.e., the list of directories in which the system searches for programs). Thus, it is usually necessary to type the full path of the command as shown above rather than just the name of the command itself.

The default runlevel for a system is specified in the /etc/inittab file, which will contain an entry such as id:3:initdefault: if the system starts in runlevel 3, or id:5:initdefault: if it starts in runlevel 5. This file can be easily (and safely) read with a command such as cat, i.e.,

```
cat /etc/inittab
```

As an alternative to telinit, the runlevel into which the system boots can be changed by modifying /etc/inittab manually with a text editor. However, it is generally easier and safer (i.e., less chance of accidental damage to the file) to use telinit. It is always wise to make a backup copy of /etc/inittab or any other configuration file before attempting to modify it manually.

# Partitioning a Disk

## Linux

```
fdisk /dev/emcpowerp OR fdisk /dev/sdb
m → n → p → 1 → enter → enter → w
```

Then format the new partition

```
mkfs -t ext2 /dev/sdb1
```

OR

```
mkfs.ext3 /dev/emcpower# OR mkfs.ext3 /dev/sdb1
```

## Mount Disk Partitions:

- ✓ First make sure the slice has no data

**Create directories that will be mounted on a slice**

e.g:

```
mkdir /rocket
cd /rocket

mkdir IFMX_ROCKET
mkdir ROCKET_DATA
```

**Mount slice to the directory**

e.g:

Linux

```
mount /dev/sdb1 /rocket/ROCKET_DATA
mount /dev/sdb2 /rocket/IFMX_ROCKET
```

**Add these entries to /etc/fstab file so the system can mount on boot up**

```
cp /etc/fstab /etc/fstab.bak
```

vi /etc/fstab and add the following lines

```
/dev/sdb1 /rocket/ROCKET_DATA ext4 defaults 1 1
/dev/sdb2 /rocket/IFMX_ROCKET ext4 defaults 1 1
```

- fdisk /dev/sdc
  - n
  - p
  - Enter for first sector
  - Enter for last sector
  - p = print the partition table
  - t = change a partition's system id
  - L = type L to list all codes
  - 8e = Partition type from Linux to Linux LVM
  - w
- 
- Create Physical Volume (PV) = pvcreate /dev/sdc1
  - Verify physical volume = pvdisplay
- 
- Create Volume Group (VG) = vgcreate oracle\_vg /dev/sdc1
  - Verify Volume group = vgdisplay oracle\_vg
- 
- Create Logical Volumes (LV) = lvcreate -n oracle\_lv -size 2G oracle\_vg
  - Verify logical volumes = lvdisplay
- 
- Format Logical Volumes = mkfs.xfs /dev/oracle\_vg/oracle\_lv
- 
- Create a new directory = mkdir /oracle
- 
- Mount the new file system = mount /dev/oracle\_vg/oracle\_lv /oracle
- 
- Verify = df -h

## To extend filesystem of a Linux VM using LVM

### Go to your virtualization product (VMWare or Oracle Virtual Box)

- Increase the disk space to desired number and then click ok

Now go to your Linux VM

- Reboot the VM to have the system re-scan the newly added disk Or
- `cd /sys/class/scsi_disk/2:0:0:0`
- `echo '1' > device/rescan`
- `fdisk -l` (To make sure the disk is increased)
- Create a new partition
  - `fdisk /dev/sdc`
  - `n` (for new partition)
  - `p` (for primary partition)
  - `2` (partition number, 2 or the new partition)
  - `Enter`
  - `Enter`
  - `t` (Label the new partition)
  - `3` (Pick default value)
  - `8e` (This will make the filesystem as LVM)
  - `w` (Write)
  - `# reboot or init 6`

Note: The above procedure will create /dev/sdc2 partition

- Extend the LVM group
  - `pvdisplay` (To see which group associated with which disk)
  - `pvs` (Info about physical volumes)
  - `vgdisplay oracle_vg` (oracle\_vg is the group name or you can simply run vgdisplay)  
On vgdisplay you will notice Free PE / Size at the bottom
  - `pvcreate /dev/sdc2` (Initialize partition for use by LVM)
  - `vgextend oracle_vg /dev/sdc2` (# = whichever partition was created above)
  - Run `vgdisplay oracle_vg`  
check (Free PE / Size). The second column is the right column as free. If it is in G convert that into M. e.g. 1G = 1024M
  - `lvextend -L+1024M /dev/mapper/oracle_vg-oracle_lv`
  - `resize2fs /dev/mapper/oracle_vg-oracle_lv`
  - OR
  - `xfs_growfs /dev/mapper/oracle_vg-oracle_lv`

## **Use a File for Additional Swap Space.**

### **What is swap? – CentOS.org**

Swap space in Linux is used when the amount of physical memory (RAM) is full. If the system needs more memory resources and the RAM is full, inactive pages in memory are moved to the swap space. While swap space can help machines with a small amount of RAM, it should not be considered a replacement for more RAM. Swap space is located on hard drives, which have a slower access time than physical memory

### **Recommended swap size = Twice the size of RAM**

Lets say,

M = Amount of RAM in GB, and S = Amount of swap in GB, then

If  $M < 2$

then  $S = M * 2$

Else  $S = M + 2$

### **Commands**

dd

mkswap

swapon or swapoff

### **Steps to Create Swap Space from Existing Disk.**

If you don't have any additional disks, you can create a file somewhere on your filesystem, and use that file for swap space.

The following dd command example creates a swap file with the name “newswap” under / directory with a size of 1024MB (1.0GB).

```
dd if=/dev/zero of=/newswap bs=1M count=1024
```

Where

if = read from FILE instead of stdin

of = write to FILE instead of stdout

bs = read and write BYTES at a time

count = total size of the file

Change the permission of the swap file so that only root can access it.

```
chmod go-r /newswap OR
chmod 0600 /newswap
```

Make this file as a swap file using mkswap command.

```
mkswap /newswap
```

Enable the newly created newswap.

```
swapon /newswap
```

To make this swap file available as a swap area even after the reboot, add the following line to the /etc/fstab file.

```
cat /etc/fstab
/newswap swap swap defaults 0 0
```

Verify whether the newly created swap area is available for your use.

```
swapon -s
free -h
```

If you don't want to reboot to verify whether the system takes all the swap space mentioned in the /etc/fstab, you can do the following, which will disable and enable all the swap partition mentioned in the /etc/fstab

```
swapoff -a
swapon -a
```

# Overview of systemd for RHEL 7

The systemd system and service manager is responsible for controlling how services are started, stopped and otherwise managed on Red Hat Enterprise Linux 7 systems. By offering on-demand service start-up and better transactional dependency controls, systemd dramatically reduces start up times. As a systemd user, you can prioritize critical services over less important services.

Although the systemd process replaces the init process (quite literally, /sbin/init is now a symbolic link to /usr/lib/systemd/systemd) for starting services at boot time and changing runlevels, systemd provides much more control than the init process does while still supporting existing init scripts. Here are some examples of the features of systemd:

- **Logging:** From the moment that the initial RAM disk is mounted to start the Linux kernel to final shutdown of the system, all log messages are stored by the new systemd journal. Before the systemd journal existed, initial boot messages were lost, requiring that you try to watch the screen as messages scrolled by to debug boot problems. Now, all system messages come in on a single stream and are stored in the **/run** directory. Messages can then be consumed by the rsyslog facility (and redirected to traditional log files in the **/var/log** directory or to remote log servers) or displayed using the **journctl** command across a variety of attributes.
- **Dependencies:** With systemd, an explicit set of dependencies can be defined for each service, instead of being implied by boot order. This allows a service to start at any point that its dependencies are met. In this way, many services can start at the same time, making the boot process faster. Likewise, complex sets of dependencies can be set up, so the exact requirements of a service (such as storage availability or file system checking) can be met before a service starts.
- **Cgroups:** Services are identified by Cgroups, which allow every component of a service to be managed. For example, the older System V init scripts would start a service by launching a process which itself might start other child processes. When the service was killed, it was hoped that the parent process would do the right thing and kill its children. By using Cgroups, all components of a service have a tag that can be used to make sure that all of those components are properly started or stopped.
- **Activating services:** Services don't just have to be always running or not running based on runlevel, as they were previous to systemd. Services can now be activated based on path, socket, bus, timer, or hardware activation. Likewise, because systemd can set up sockets, if a process handling communications goes away, the process that starts up in its place can pick up the next message from the socket. To the clients using the service, it can look as though the service continued without interruption.
- **More than services:** Instead of just managing services, systemd can manage several different unit types. These unit types include:
  - **Devices:** Create and use devices.
  - **Mounts and automounts:** Mount file systems upon request or automount a file system based on a request for a file or directory within that file system.
  - **Paths:** Check the existence of files or directories or create them as needed.

- **Services:** Start a service, which often means launching a service daemon and related components.
- **Slices:** Divide up computer resources (such as CPU and memory) and apply them to selected units.
- **Snapshots:** Take snapshots of the current state of the system.
- **Sockets:** Set up sockets to allow communication paths to processes that can remain in place, even if the underlying process needs to restart.
- **Swaps:** Create and use swap files or swap partitions.
- **Targets:** Manage a set of services under a single unit, represented by a target name rather than a runlevel number.
- **Timers:** Trigger actions based on a timer.
- **Resource management**
  - The fact that each systemd unit is always associated with its own cgroup lets you control the amount of resources each service can use. For example, you can set a percent of CPU usage by service which can put a cap on the total amount of CPU that service can use -- in other words, spinning off more processes won't allow more resources to be consumed by the service. Prior to systemd, *nice* levels were often used to prevent processes from hogging precious CPU time. With systemd's use of cgroups, precise limits can be set on CPU and memory usage, as well as other resources.
  - A feature called *slices* lets you slice up many different types of system resources and assign them to users, services, virtual machines, and other units. Accounting is also done on these resources, which can allow you to charge customers for their resource usage.

## Booting RHEL 7 with systemd

When you boot a standard X86 computer to run RHEL 7, the BIOS boots from the selected medium (usually a local hard disk) and the boot loader (GRUB2 for RHEL 7) starts the RHEL 7 kernel and initial RAM disk. After that, the systemd process takes over to initialize the system and start all the system services.

Although there is not a strict order in which services are started when a RHEL 7 (systemd) system is booted, there is a structure to the boot process. The direction that the systemd process takes at boot time depends on the **default.target** file. A long listing of the **default.target** file shows you which target starts when the system boots:

```
cd /etc/systemd/system
ls -l default.target
lrwxrwxrwx. 1 root root 16 Aug 23 19:18 default.target ->
/lib/systemd/system/graphical.target
```

You can see here that the **graphical.target** (common for desktop systems or servers with graphical interfaces) is set as the **default.target** (via a symbolic link). To understand what targets, services and other units start up with the graphical target, it helps to work backwards, as systemd does, to build the dependency tree. Here's what to look for:

- **graphical.target**: The **/lib/systemd/system/graphical.target** file includes these lines:
 

```
Requires=multi-user.target
Wants=display-manager.service
Conflicts=rescue.service rescue.target
After=multi-user.target rescue.service rescue.target display-
manager.service
AllowIsolate=yes
```

This tells systemd to start everything in the multi-user.target before starting the graphical target. Once that's done, the "Wants" entry tells systemd to start the **display-manager.service** service (**/etc/systemd/system/display-manager.service**), which runs the GNOME display manager (**/usr/sbin/gdm**).

- **multi-user.target**: The **/usr/lib/systemd/system/multi-user.target** starts the services you would expect in a RHEL multi-user mode. The file contains the following line:

```
Requires=basic.target
```

This tells systemd to start everything in the **/usr/lib/systemd/system/basic.target** target before starting the other multi-user services. After that, for the multi-user.target, all units (services, targets, etc.) in the **/etc/systemd/system/multi-user.target.wants** and **/usr/lib/systemd/system/multi-user.target.wants** directories are started. When you enable a service, a symbolic link is placed in the **/etc/systemd/system/multi-user.target.wants** directory. That directory is where you will find links to most of the services you think of as starting in multi-user mode (printing, cron, auditing, SSH, and so on). Here is an example of the services, paths, and targets in a typical **multi-user.target.wants** directory:

```
cd /etc/systemd/system/multi-user.target.wants
abrt-ccpp.service hypervkvpd.service postfix.service
abrtd.service hypervvssd.service remote-fs.target
abrt-oops.service irqbalance.service rhsmcertd.service
abrt-vmcore.service ksm.service rngd.service
abrt-xorg.service ksmtuned.service rpcbind.service
atd.service libstoragemgmt.service rsyslog.service
auditd.service libvirtd.service smartd.service
avahi-daemon.service mdmonitor.service sshd.service
chronyd.service ModemManager.service sysstat.service
crond.service netcf-transaction.service tuned.service
cups.path nfs.target vmtoolsd.service
```

- **basic.target**: The **/usr/lib/systemd/system/basic.target** file starts the basic services associated with all running RHEL 7 systems. The file contains the following line:

```
Requires=sysinit.target
```

This points systemd to the **/usr/lib/systemd/system/sysinit.target**, which must start before the **basic.target** can continue. The **basic.target** target file starts the firewalld and microcode services from the **/etc/systemd/system/basic.target.wants** directory and

services for SELinux, kernel messages, and loading modules from the **/usr/lib/systemd/system/basic.target.wants** directory.

- **sysinit.target**: The **/usr/lib/systemd/system/sysinit.target** file starts system initialization services, such as mounting file systems and enabling swap devices. The file contains the following line:

```
Wants=local-fs.target swap.target
```

Besides mounting file systems and enabling swap devices, the sysinit.target starts targets, services, and mounts based on units contained in the **/usr/lib/systemd/system/sysinit.target.wants** directory. These units enable logging, set kernel options, start the udevd daemon to detect hardware, and allow file system decryption, among other things. The **/etc/systemd/system/sysinit.target.wants** directory contains services that start iSCSI, multipath, LVM monitoring and RAID services.

- **local-fs.target**: The local-fs.target is set to run after the local-fs-pre.target target, based on this line:

```
After=local-fs-pre.target
```

There are no services associated with the local-fs-pre.target target (you could add some to a "wants" directory if you like). However, units in the **/usr/lib/systemd/system/local-fs.target.wants** directory import the network configuration from the initramfs, run a file system check (fsck) on the root file system when necessary, and remounting the root file system (and special kernel file systems) based on the contents of the /etc/fstab file.

Although the boot process is built by systemd in the order just shown, it actually runs, in general, in the opposite order. As a rule, a target on which another target is dependent must be running before the units in the first target can start. To see more details about the boot process, see the bootup man page (**man 7 bootup**).

## Using the **systemctl** Command

The most important command for managing services on a RHEL 7 (systemd) system is the **systemctl** command. Here are some examples of the **systemctl** command (using the nfs-server service as an example) and a few other commands that you may find useful:

- **Checking service status**: To check the status of a service (for example, nfs-server.service), type the following:
  - **# systemctl status nfs-server.service**
  - nfs-server.service - NFS Server
  - Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; disabled)
  - Active: active (exited) since Wed 2014-03-19 10:29:40 MDT; 57s ago

- Process: 5206 ExecStartPost=/usr/libexec/nfs-utils/scripts/nfs-server.postconfig (code=exited, status=0/SUCCESS)
- Process: 5191 ExecStart=/usr/sbin/rpc.nfsd \$RPCNFSDARGS \$RPCNFSDCOUNT (code=exited, status=0/SUCCESS)
- Process: 5188 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)
- Process: 5187 ExecStartPre=/usr/libexec/nfs-utils/scripts/nfs-server.preconfig (code=exited, status=0/SUCCESS)
- Main PID: 5191 (code=exited, status=0/SUCCESS)
- CGroup: /system.slice/nfs-server.service
- 
- Mar 19 10:29:40 localhost.localdomain systemd[1]: Starting NFS Server...
- Mar 19 10:29:40 localhost.localdomain systemd[1]: Started NFS Server.
- **Stopping a service:** To stop a service, use the stop option as follows:  
• # **systemctl stop nfs-server.service**
- **Starting a service:** To start a service, use the start option as follows:  
• # **systemctl start nfs-server.service**
- **Enabling a service:** To enable a service so it starts automatically at boot time, type the following:  
• # **systemctl enable nfs-server.service**
- **Disable a service:** To disable a service so it doesn't start automatically at boot time, type the following:  
• # **systemctl disable nfs-server.service**
- **Listing dependencies:** To see dependencies of a service, use the list-dependencies option, as follows:  
• # **systemctl list-dependencies nfs-server.service**
  - nfs-server.service
    - |nfs-idmap.service
    - |nfs-mountd.service
    - |nfs-rquotad.service
    - |proc-fs-nfsd.mount
    - |rpcbind.service
    - |system.slice
    - |var-lib-nfs-rpc\_pipefs.mount
    - |basic.target
      - |alsa-restore.service
      - |alsa-state.service
    - ...
- **Listing units in targets:** To see what services and other units (service, mount, path, socket, and so on) are associated with a particular target, type the following:  
• # **systemctl list-dependencies multi-user.target**
  - multi-user.target
    - |abrt-ccpp.service
    - |abrt-oops.service
    - |abrt-vmcore.service
    - |abrt-xorg.service
    - |abrttd.service
    - |atd.service

- auditd.service
- avahi-daemon.service
- brandbot.path
- chronyd.service
- crond.service
- ...
- **List specific types of units:** Use the following command to list specific types of units (in these examples, service and mount unit types):
 

```
systemctl list-units --type service
```

UNIT	LOAD	ACTIVE	SUB	DESCRIPTION
abrt-ccpp.service	loaded	active	exited	Install ABRT coredump hook
abrt-oops.service		loaded	active	running ABRT kernel log watcher
abrt-xorg.service		loaded	active	running ABRT Xorg log watcher
abrtd.service		loaded	active	running ABRT Automated Bug Reporting
accounts-daemon.service		loaded	active	running Accounts Service
...				

```
systemctl list-units --type mount
```

UNIT	LOAD	ACTIVE	SUB	DESCRIPTION
-.mount	loaded	active	mounted	/
boot.mount	loaded	active	mounted	/boot
dev-hugepages.mount	loaded	active	mounted	Huge Pages File System
dev-mqueue.mount	loaded	active	mounted	POSIX Message Queue File Syst
mnt-repo.mount	loaded	active	mounted	/mnt/repo
proc-fs-nfsd.mount	loaded	active	mounted	RPC Pipe File System
run-user-1000-gvfs.mount	loaded	active	mounted	/run/user/1000/gvfs
...				
- **Listing all units:** To list all units installed on the system, along with their current states, type the following:
 

```
systemctl list-unit-files
```

UNIT FILE	STATE
proc-sys-fs-binfmt_misc.automount	static
dev-hugepages.mount	static
dev-mqueue.mount	static
proc-sys-fs-binfmt_misc.mount	static
...	
arp-ethers.service	disabled
atd.service	enabled
auditd.service	enabled
...	
- **View service processes with systemd-cgtop:** To view processes associated with a particular service (cgroup), you can use the **systemd-cgtop** command. Like the **top** command (which sorts processes by such things as CPU and memory usage), **systemd-**

**cgtop** lists running processes based on their service (cgroup label). Once systemd-cgtop is running, you can press keys to sort by memory (m), CPU (c), task (t), path (p), or I/O load (i). Here is an example:

- `# systemd-cgtop`
- **Recursively view cgroup contents:** To output a recursive list of cgroup content, use the `systemd-cgls` command:

```
systemd-cgls
└─user.slice
 └─user-1000.slice
 └─session-5.scope
 ├─2661 gdm-session-worker [pam/gdm-password]
 ├─2672 /usr/bin/gnome-keyring-daemon --daemonize --login
 ├─2674 gnome-session --session gnome-classic
 ├─2682 dbus-launch --sh-syntax --exit-with-session
 ├─2683 /bin/dbus-daemon --fork --print-pid 4 --print-address 6 --
 session
 ├─2748 /usr/libexec/gvfsd
...

```
- **View journal (log) files:** Using the `journalctl` command you can view messages from the systemd journal. Using different options you can select which group of messages to display. The `journalctl` command also supports tab completion to fill in fields for which to search. Here are some examples:
  - `# journalctl -h` *View help for the command*
  - `# journalctl -k` *View kernel messages from current boot*
  - `# journalctl -f` *Follow journal messages (like tail -f)*
  - `# journalctl -u NetworkManager` *View messages for specific unit (can tab complete)*

## Comparing systemd to Traditional init

Some of the benefits of systemd over the traditional System V init facility include:

- systemd never loses initial log messages
- systemd can respawn daemons as needed
- systemd records runtime data (i.e., captures stdout/stderr of processes)
- systemd doesn't lose daemon context during runtime
- systemd can kill all components of a service cleanly

Here are some details of how systemd compares to pre-RHEL 7 init and related commands:

- **System startup:** The `systemd` process is the first process ID (PID 1) to run on RHEL 7 system. It initializes the system and launches all the services that were once started by the traditional `init` process.
- **Managing system services:** For RHEL 7, the `systemctl` command replaces `service` and `chkconfig`. Prior to RHEL 7, once RHEL was up and running, the `service` command was used to start and stop services immediately. The `chkconfig` command was used to identify at which run levels a service would start or stop automatically.

Although you can still use the **service** and **chkconfig** commands to start/stop and enable/disable services, respectively, they are not 100% compatible with the RHEL 7 **systemctl** command. For example, non-standard service options, such as those that start databases or check configuration files, may not be supported in the same way for RHEL 7 services.

- **Changing runlevels:** Prior to RHEL 7, runlevels were used to identify a set of services that would start or stop when that runlevel was requested. Instead of runlevels, systemd uses the concept of *targets* to group together sets of services that are started or stopped. A target can also include other targets (for example, the multi-user target includes an nfs target).

There are systemd targets that align with the earlier runlevels. However the point of targets is not to necessarily imply a level of activity (for example, runlevel 3 implied more services were active than runlevel 1). Instead targets just represent a group of services, so it's appropriate that there are many more targets available than there are runlevels. The following list shows how systemd targets align with traditional runlevels:

- Traditional runlevel      New target name      Symbolically linked to...
  - Runlevel 0                    |      runlevel0.target -> poweroff.target
  - Runlevel 1                    |      runlevel1.target -> rescue.target
  - Runlevel 2                    |      runlevel2.target -> multi-user.target
  - Runlevel 3                    |      runlevel3.target -> multi-user.target
  - Runlevel 4                    |      runlevel4.target -> multi-user.target
  - Runlevel 5                    |      runlevel5.target -> graphical.target
  - Runlevel 6                    |      runlevel6.target -> reboot.target
- **Default runlevel:** The default runlevel (previously set in the **/etc/inittab** file) is now replaced by a default target. The location of the default target is **/etc/systemd/system/default.target**, which by default is linked to the multi-user target.
- **Location of services:** Before systemd, services were stored as scripts in the **/etc/init.d** directory, then linked to different runlevel directories (such as **/etc/rc3.d**, **/etc/rc5.d**, and so on). Services with systemd are named *something.service*, such as **firewalld.service**, and are stored in **/lib/systemd/system** and **/etc/systemd/system** directories. Think of the **/lib** files as being more permanent and the **/etc** files as the place you can modify configurations as needed.

When you enable a service in RHEL 7, the service file is linked to a file in the **/etc/systemd/system/multi-user.target.wants** directory. For example, if you run **systemctl enable fcoe.service** a symbolic link is created from **/etc/systemd/system/multi-user.target.wants/fcoe.service** that points to **/lib/systemd/system/fcoe.service** to cause the **fcoe.service** to start at boot time.

Also, the older System V init scripts were actual shell scripts. The systemd files tasked to do the same job are more like .ini files that contain the information needed to launch a service.

- **Configuration files:** The **/etc/inittab** file was used by the init process in RHEL 6 and earlier to point to the initialization files (such as **/etc/rc.sysinit**) and runlevel service directories (such as **/etc/rc5.d**) needed to start up the system. Changes to those services was done in files (usually named after the service) in the **/etc/sysconfig** directory. For systemd in RHEL 7, there are still files in **/etc/sysconfig** used to modify how services behave. However, services can be modified by adding files to the **/etc/systemd** directory to override the permanent service files in the **/lib/systemd** directories.

# Transitioning to systemd

If you are used to using the **init** process and System V init scripts prior to RHEL 7, there are a few things you should know about transitioning to systemd:

- **Using RHEL 6 commands:** For the time being, you can use commands such as **service**, **chkconfig**, **runlevel**, and **init** as you did in RHEL 6. They will cause appropriate systemd commands to run, with similar, if not exactly the same, results. Here are some examples:
  - `# service cups restart`
  - Redirecting to `/bin/systemctl restart cups.service`
  - `# chkconfig cups on`
  - Note: Forwarding request to '`systemctl enable cups.service`'.
- **System V init Scripts:** Although not encouraged, System V init scripts are still supported. There are still some services in RHEL 7 that are implemented in System V init scripts. To see System V init scripts that are available on your system and the runlevels on which they start, use the **chkconfig** command as follows:
  - `# chkconfig --list`
  - `...`
  - `iprdump 0:off 1:off 2:on 3:on 4:on 5:on 6:off`
  - `iprinit 0:off 1:off 2:on 3:on 4:on 5:on 6:off`
  - `iprupdate 0:off 1:off 2:on 3:on 4:on 5:on 6:off`
  - `netconsole 0:off 1:off 2:off 3:off 4:off 5:on 6:off`
  - `network 0:off 1:off 2:on 3:on 4:on 5:on 6:off`
  - `rhnasd 0:off 1:off 2:on 3:on 4:on 5:on 6:off`
  - `...`

Using **chkconfig**, however, will not show you the whole list of services on your system. To see the systemd-specific services, run the **systemctl list-unit-files** command, as described earlier.

## Customizing motd

You can have the MOTD (message of the day) display messages that may be unique to the machine. One way to do this is to create a script that runs when a user logs on to the system.

First, create a script in **/etc/profile.d = touch motd.sh**

Make it executable = **chmod a+x motd.sh** (make sure it has the extension as .sh)

```
#!/bin/bash
#
echo -e "
#####
Welcome to `hostname`
This system is running `cat /etc/redhat-release`
kernel is `uname -r`
#
You are logged in as `whoami`
#
#####"
"
```

Next, edit **/etc/ssh/sshd\_config** as follows:

```
PrintMotd no
```

This will disable motd

Now restart the sshd service

```
systemctl restart sshd.service
```

That's it! When you log in, you'd see something similar to:

```
#####

Welcome to MyFirstLinuxVM
This system is running CentOS Linux release 7.5.1804 (Core)
kernel is 3.10.0-862.el7.x86_64

You are logged in as iafzal

#####
```

### Steps for NFS Server Configuration

- Install NFS packages

```
yum install nfs-utils libnfsidmap (most likely they are installed)
```

- Once the packages are installed, enable and start NFS services

```
systemctl enable rpcbind
```

```
systemctl enable nfs-server
```

```
systemctl start rpcbind, nfs-server, rpc-statd, nfs-idmapd
```

- Create NFS share directory and assign permissions

```
mkdir /mypretzels
```

```
chmod a+rwx /mypretzels
```

- Modify **/etc/exports** file to add new shared filesystem

```
/mypretzels 192.168.12.7(rw,sync,no_root_squash) = for only 1 host
```

```
/mypretzels *(rw,sync,no_root_squash) = for everyone
```

- Export the NFS filesystem

```
exportfs -rv
```

- Stop and disable firewalld

```
systemctl stop firewalld
```

```
systemctl disable firewalld
```

### Steps for NFS Client Configuration

- Install NFS packages

```
yum install nfs-utils rpcbind
```

- Once the packages are installed enable and start rpcbind service

```
systemctl rpcbind start
```

- Make sure firewalld or iptables stopped (if running)

```
ps -ef | egrep "firewall|iptable"
```

- Show mount from the NFS server

```
showmount -e 192.168.1.5 (NFS Server IP)
 • Create a mount point

mkdir /mnt/kramer
 • Mount the NFS filesystem

mount 192.168.1.5:/mypretzels /mnt/kramer
 • Verify mounted filesystem

df -h
 • To unmount

umount /mnt/kramer
```

## Samba download, install and configuration

- Samba is a Linux tool or utility that allows sharing for Linux resources such as files and printers to with other operating systems
- It works exactly like NFS but the difference is NFS shares within Linux or Unix like system whereas Samba shares with other OS (e.g. Windows, MAC etc.)

For example, computer “A” shares its filesystem with computer “B” using Samba then computer “B” will see that shared filesystem as if it is mounted as the local filesystem

- Samba shares its filesystem through a protocol called **SMB** (Server Message Block) which was invented by IBM
- Another protocol used to share Samba is through **CIFS** (Common Internet File System) invented by Microsoft and also NMB (NetBios Name server)
- **CIFS** became the extension of **SMB** and now Microsoft has introduced newer version of SMB v2 and v3 that are mostly used in the industry
- Most people, when they use either SMB or CIFS, are talking about the same exact thing. The two are interchangeable not only in discussion, but also in application – i.e., a client speaking CIFS can talk to a server speaking SMB and vice versa. Why? Because CIFS is a form of SMB

### Step by steps installation instructions

First please make sure to take a snapshot of your VM

- Install samba packages

```
Become root user
yum install samba samba-client samba-common
```
- Enable samba to be allowed through firewall (Only if you have firewall running)

```
firewall-cmd --permanent --zone=public --add-service=samba
firewall-cmd -reload
```
- To stop and disable firewall or iptables

```
systemctl stop firewalld
systemctl stop iptables
systemctl disable firewalld
systemctl disable iptables
```

- Create Samba share directory and assign permissions
 

```
mkdir -p /samba/morepretzels
chmod a+rwx /samba/morepretzels
chown -R nobody:nobody /samba
```
- Also, you need to change the SELinux security context for the samba shared directory as follows: (*Only if you have SELinux enabled*)
 

```
chcon -t samba_share_t /samba/morepretzels
```
- If you want to disable SELinux, follow these instructions
 

```
sestatus To check the SELinux status)
vi /etc/selinux/config
```

 Change
 

```
SELINUX=enforcing
```

 To
 

```
SELINUX=disabled
```

 # reboot
- Modify `/etc/samba/smb.conf` file to add new shared filesystem (*Make sure to create a copy of smb.conf file*)
 Delete everything from smb.conf file and add the following parameters

```
[global]
 workgroup = WORKGROUP
 netbios name = centos
 security = user
 map to guest = bad user
 dns proxy = no

[Anonymous]
 path = /samba/morepretzels
 browsable = yes
 writable = yes
 guest ok = yes
 guest only = yes
 read only = no
```

- Verify the setting
 

```
testparm
```
- Once the packages are installed, enable and start Samba services
 

```
systemctl enable smb
systemctl enable nmb
systemctl start smb
systemctl start nmb
```

- Mount on Windows client
  - Go to start
  - Go to search bar
  - Type [\\192.168.1.95](http://192.168.1.95) (This is my server IP, you can check your Linux CentOS IP by running the command ifconfig)

- Mount on Linux client

Become root

```
yum -y install cifs-utils samba-client
Create a mount point directory
mkdir /mnt/sambashare
Mount the samba share
mount -t cifs //192.168.1.95/Anonymous /mnt/sambashare/
Entry without password
```

## Secure Samba Server

- Create a group smbgrp & user larry to access the samba server with proper authentication

```
useradd larry
groupadd smbgrp
usermod -a -G smbgrp larry
smbpasswd -a larry
New SMB password: YOUR SAMBA PASS
Retype new SMB password: REPEAT YOUR SAMBA PASS
Added user larry
```

- Create a new share, set the permission on the share:

```
mkdir /samba/securepretzels
chown -R larry:smbgrp /samba/securepretzels
chmod -R 0770 /samba/securepretzels
chcon -t samba_share_t /samba/securepretzels
```

- Edit the configuration file /etc/samba/smb.conf (Create a backup copy first)

```
vi /etc/samba/smb.conf
Add the following lines
[Secure]
 path = /samba/securepretzels
 valid users = @smbgrp
 guest ok = no
```

```
writable = yes
Browsable = yes
```

- Restart the services
  - # systemctl restart smb
  - # systemctl restart nmb

## **Linux Boot Sequence**

### **1. BIOS**

- BIOS stands for Basic Input/Output System
- Performs some system integrity checks
- Searches, loads, and executes the boot loader program.
- It looks for boot loader in floppy, cd-rom, or hard drive. You can press a key (typically F12 or F2, but it depends on your system) during the BIOS startup to change the boot sequence.
- Once the boot loader program is detected and loaded into the memory, BIOS gives the control to it.
- So, in simple terms BIOS loads and executes the MBR boot loader.

### **2. MBR**

- MBR stands for Master Boot Record.
- It is located in the 1st sector of the bootable disk. Typically /dev/hda, or /dev/sda
- MBR is less than 512 bytes in size. This has three components 1) primary boot loader info in 1st 446 bytes 2) partition table info in next 64 bytes 3) mbr validation check in last 2 bytes.
- It contains information about GRUB (or LILO in old systems).
- So, in simple terms MBR loads and executes the GRUB boot loader.

### **3. GRUB**

- GRUB stands for Grand Unified Bootloader.
- If you have multiple kernel images installed on your system, you can choose which one to be executed.
- GRUB displays a splash screen, waits for few seconds, if you don't enter anything, it loads the default kernel image as specified in the grub configuration file.
- GRUB has the knowledge of the filesystem (the older Linux loader LILO didn't understand filesystem).
- Grub configuration file is /boot/grub/grub.conf (/etc/grub.conf is a link to this). The following is sample grub.conf of CentOS.

```
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.18-194.el5PAE)
 root (hd0,0)
 kernel /boot/vmlinuz-2.6.18-194.el5PAE ro root=LABEL=/
 initrd /boot/initrd-2.6.18-194.el5PAE.img
```

- As you notice from the above info, it contains kernel and initrd image.
- So, in simple terms GRUB just loads and executes Kernel and initrd images.

### **4. Kernel**

- Mounts the root file system as specified in the "root=" in grub.conf
- Kernel executes the /sbin/init program
- Since init was the 1st program to be executed by Linux Kernel, it has the process id (PID) of 1. Do a 'ps -ef | grep init' and check the pid.
- initrd stands for Initial RAM Disk.

- initrd is used by kernel as temporary root file system until kernel is booted and the real root file system is mounted. It also contains necessary drivers compiled inside, which helps it to access the hard drive partitions, and other hardware.

## 5. Init

- Looks at the /etc/inittab file to decide the Linux run level.
- Following are the available run levels
  - 0 – halt
  - 1 – Single user mode
  - 2 – Multiuser, without NFS
  - 3 – Full multiuser mode
  - 4 – unused
  - 5 – X11
  - 6 – reboot
- Init identifies the default initlevel from /etc/inittab and uses that to load all appropriate program.
- Execute ‘grep initdefault /etc/inittab’ on your system to identify the default run level
- If you want to get into trouble, you can set the default run level to 0 or 6. Since you know what 0 and 6 means, probably you might not do that.
- Typically you would set the default run level to either 3 or 5.

## 6. Runlevel programs

- When the Linux system is booting up, you might see various services getting started. For example, it might say “starting sendmail .... OK”. Those are the runlevel programs, executed from the run level directory as defined by your run level.
- Depending on your default init level setting, the system will execute the programs from one of the following directories.
  - Run level 0 – /etc/rc.d/rc0.d/
  - Run level 1 – /etc/rc.d/rc1.d/
  - Run level 2 – /etc/rc.d/rc2.d/
  - Run level 3 – /etc/rc.d/rc3.d/
  - Run level 4 – /etc/rc.d/rc4.d/
  - Run level 5 – /etc/rc.d/rc5.d/
  - Run level 6 – /etc/rc.d/rc6.d/
- Please note that there are also symbolic links available for these directory under /etc directly. So, /etc/rc0.d is linked to /etc/rc.d/rc0.d.
- Under the /etc/rc.d/rc\*.d/ directories, you would see programs that start with S and K.
- Programs starts with S are used during startup. S for startup.
- Programs starts with K are used during shutdown. K for kill.
- There are numbers right next to S and K in the program names. Those are the sequence number in which the programs should be started or killed.
- For example, S12syslog is to start the syslog deamon, which has the sequence number of 12. S80sendmail is to start the sendmail daemon, which has the sequence number of 80. So, syslog program will be started before sendmail.

# RED HAT ENTERPRISE LINUX 5, 6, AND 7

## Common administrative commands

	RHEL5	RHEL6	RHEL7	
<b>SYSTEM BASICS</b>				
View subscription information	/etc/sysconfig/rhn/systemid	/etc/sysconfig/rhn/systemid subscription-manager identity	subscription-manager identity	
Configure subscription	rhn_register	rhn_register rhnreg_ks subscription-manager	subscription-manager <sup>1</sup> rhn_register <sup>2</sup>	
View system profile	sosreport dmidecode hwbrowser		sosreport dmidecode lshw	
View RHEL version information			/etc/redhat-release	
	1 subscription-manager is used for Satellite 6, Satellite 5.6 with SAM and newer, and Red Hat's CDN. 2 RHN tools are deprecated on Red Hat Enterprise Linux 7. rhn_register should be used for Satellite server 5.6 and newer only. For details, see: <a href="#">Satellite 5.6 unable to register RHEL 7 client system due to rhn-setup package not included in Minimal Installation</a>			
<b>BASIC CONFIGURATION</b>				
Graphical configuration tools		system-config-*	gnome-control-center	
Configure network		system-config-network	nmcli nm-tui nm-connection-editor	
Configure system language		system-config-language	localectl	
Configure time and date		system-config-date	timedatectl date	
Synchronize time and date	ntpdate /etc/ntp.conf		timedatectl /etc/chrony.conf	
Configure keyboard		system-config-keyboard	localectl	
Text-based configuration tools		system-config-* tui		
Configure printer		system-config-printer		
Configure samba		smbclient /etc/samba/smb.conf smbpasswd		
Configure SSH		/etc/ssh/ssh_config /etc/ssh/sshd_config -/.ssh/config ssh-keygen		
<b>JOBs AND SERVICES</b>				
Configure logging	/etc/syslog.conf	/etc/rsyslog.conf	/etc/rsyslog.conf /etc/rsyslog.d/*.*.conf /var/log/journal systemd-journal.service	
List all services		chkconfig --list ls /etc/init.d/	systemctl -at service ls /etc/systemd/system/*.service ls /usr/lib/systemd/system/*.service	
List running services		service --status-all	systemctl -t service --state=active	
Start/stop service		service name start service name stop	systemctl start name.service systemctl stop name.service	
Enable/disable service		chkconfig name on chkconfig name off	systemctl enable name.service systemctl disable name.service	
View service status		service name status	systemctl status name.service	
Check if service is enabled		chkconfig name	systemctl is-enabled name	
Create new service file or modify configuration		chkconfig --add	systemctl daemon-reload	
View run level/target		runlevel who -r	systemctl get-default who -r	
Change run level/target		/etc/inittab init run_level	systemctl isolate name.target systemctl set-default	
View logs		/var/log	/var/log journalctl	
Configure system audit		add audit=1 to kernel cmdline audictl /etc/audit/auditd.conf /etc/audit/audit.rules authconfig /etc/pam.d/system-auth pam_tty_audit kernel module		
View audit output		aureport /var/log/faillog		
Schedule tasks		cron at		
Configure batch tasks		batch		
Find file by name		locate		
Find file by characteristic		find		
Create archive		tar cpio zip gzip bzip2		
<b>KERNEL, BOOT, AND HARDWARE</b>				
Single user/rescue mode	append 1 or s or init=/bin/bash to kernel cmdline		append rd.break or init=/bin/bash to kernel cmdline	
Shut down system	shutdown		systemctl shutdown	
Power off system	poweroff		systemctl poweroff	
Halt system	halt		systemctl halt	
Reboot system	reboot		systemctl reboot	
Configure default run level/target	/etc/inittab		systemctl set-default	
Configure GRUB bootloader	/boot/grub/grub.conf		/etc/default/grub grub2-mkconfig grub-set-default	
View hardware configured	hwbrowser	lshw		
Configure kernel module		modprobe		
Configure hardware device		udev		
View kernel parameters		sysctl -a cat /proc/cmdline		
Load kernel module		modprobe		
Remove kernel module		modprobe -r		
View kernel version		rpm -q kernel uname -r		
<b>SOFTWARE MANAGEMENT</b>				
Install software		yum install yum groupinstall	yum install yum group install	
View software info		yum info yum groupinfo	yum info yum group info	
Update software		yum update		
Upgrade software		yum upgrade		
Configure software repository		/etc/yum.repos.d/*.repo		
Find file in package		rpm -qf filename		
View software version		rpm -q packagename		
View installed software		rpm -qa		
<b>SECURITY AND IDENTITY</b>				
Configure system security		/etc/selinux/config chcon restorecon semanage setsebool system-config-selinux		
Report on system security		sealert		
LDAP, SSSD, Kerberos		authconfig authconfig-tui authconfig-gtk		
Network users		getent		
<b>FILE SYSTEMS, VOLUMES, AND DISKS</b>				
Default file system		ext3	ext4	xfs
Defragment disk space		copy data to new file system fsck (look for 'non-contiguous inodes')	copy data to new file system fsck (look for 'non-contiguous inodes') xfs_fsr	
Create/modify disk partitions		fdisk parted	fdisk gdisk parted ssm create	
Format disk partition		mkfs.filesystem_type (ext4, xfs) mkswap	mkfs.filesystem_type (ext4, xfs) mkswap ssm create	
Mount storage		mount /etc/fstab	mount /etc/fstab ssm mount	
Create physical volume		pvcreate	pvcreate ssm create (if backend is lvm)	
Create volume group		vgcreate	vgcreate ssm create (if backend is lvm)	
Create logical volume		lvcreate	lvcreate ssm create (if backend is lvm)	
Enlarge volumes formatted with default file system		vgextend lvextend resize2fs	vgextend lvextend resize2fs	xvfs cannot currently be shrunk; copy desired data to a smaller file system.
Shrink volumes formatted with default file system		resize2fs lvreduce vgreduce		
Check/repair file system		fsck	fsck ssm check	
Configure NFS share		/etc/exports service nfs reload	/etc/exports systemctl reload nfs.service	
Mount and activate swap			/etc/fstab swapon -a	
Automatically mount at boot			/etc/fstab	
View free disk space			df	
View logical volume info			lvdisplay lvs vgdisplay vgs pvdisk pvs	
View NFS share			showmount -e mount	
Automatically mount after boot			/etc/auto.master.d/*. autofs /etc/auto.*	
Change file permissions			chmod chown chgrp umask	
Change access control list			setfacl	
<b>NETWORKING</b>				
Configure firewall		iptables and ip6tables /etc/sysconfig/iptables	iptables and ip6tables /etc/sysconfig/iptables system-config-firewall	firewall-cmd firewall-config
Configure DHCP client		/etc/dhcpd.conf /etc/dhcp6c.conf		dhcpd /etc/dhcp/dhcpd.conf /etc/sysconfig/dhcpd
Configure name resolution			/etc/hosts /etc/resolv.conf	/etc/hosts /etc/resolv.conf nmcli con mod
Configure hostname			/etc/sysconfig/network	hostnamectl /etc/hostname nmcli
View network interface info			ip addr ifconfig brctl	ip addr nmcli dev show teamdctl brctl bridge
Configure network interface			/etc/sysconfig/network-scripts/ifcfg-*	/etc/sysconfig/network-scripts/ifcfg-* nmcli con [add mod edit] nmui nm-connection-editor
View ports/sockets			ss lsof netstat	ss lsof
View routes			ip route	
Configure routes			/etc/sysconfig/network system-config-network	
<b>RESOURCE MANAGEMENT</b>				
View system usage		top ps sar iostat netstat vmstat mpstat numastat	top ps sar iostat netstat ss vmstat mpstat numastat tuna	top ps sar iostat ss vmstat mpstat numastat tuna
View disk usage		df		df iostat
Trace system calls				strace
Trace library calls				ltrace
Change process priority				nice renice
Change process run location				taskset
Kill a process				kill pkill killall

# CHEATSHEETS

## Linux Permissions Cheat Sheet

---

I created this repository in hopes that it may be used as a helpful reference.

### Permissions

---

Permissions on Unix and other systems like it are split into three classes.

- User
- Group
- Other

Files and directories are owned by a **user**.

Files and directories are also assigned to a **group**.

If a user is not the owner, nor a member of the group, then they are classified as **other**.

### Changing permissions

---

In order to change permissions, we need to first understand the two notations of permissions.

1. Symbolic notation
2. Octal notation

#### Symbolic notation

Symbolic notation is what you'd see on the left-hand side if you ran a command like `ls -l` in a terminal.

The first character in symbolic notation indicates the *file type* and isn't related to permissions in any way. The remaining characters are in sets of three, each representing a class of permissions.

The first class is the **user** class. The second class is the **group** class. The third class is the **other** class.

Each of the three characters for a class represents the read, write and execute permissions.

- r will be displayed if reading is permitted
- w will be displayed if writing is permitted
- x will be displayed if execution is permitted
- - will be displayed in the place of r, w, and x, if the respective permission is *not* permitted

Here are some examples of symbolic notation:

- -rwxr--r--: A regular file whose **user** class has read/write/execute, **group** class has only read permissions, **other** class has only read permissions
- drw-rw-r--: A directory whose **user** class has read/write permissions, **group** class has read/write permissions, **other** class has only read permissions
- crwxrw-r--: A character special file whose **user** has read/write/execute permissions, **group** class has read/write permissions, **other** class has only read permissions

### Octal notation

Octal (base-8) notation consists of at least 3 digits (sometimes 4, the left-most digit, which represents the setuid bit, the setgid bit, and the sticky bit).

Each of the three right-most digits are the sum of its component bits in the binary numeral system.

For example:

- The read bit (r in symbolic notation) adds 4 to its total
- The write bit (w in symbolic notation) adds 2 to its total
- The execute bit (x in symbolic notation) adds 1 to its total

So what number would you use if you wanted to set a permission to read and write?  $4 + 2 = 6$ .

Symbolic notation	Octal notation	Plain English
-rwxr--r--	0744	<b>user</b> class can read/write/execute; <b>group</b> class can read; <b>other</b> class can read
-rw-rw-r--	0664	<b>user</b> class can read/write; <b>group</b> class can read/write; <b>other</b> class can read
-rwxrwxr--	0774	<b>user</b> class can read/write/execute; <b>group</b> class can read/write/execute; <b>other</b> class can read
-----	0000	None of the classes have permissions
-rwx-----	0700	<b>user</b> class can read/write/execute; <b>group</b> class has no permissions; <b>other</b> class has no permissions
-rwxrwxrwx	0777	All classes can read/write/execute
-rw-rw-rw	0666	All classes can read/write
-r-xr-xr-x	0555	All classes can read/execute
-r--r--r--	0444	All classes can read

--wx-wx-wx	0333	All classes can write/execute
--w--w--w-	0222	All classes can write
---x--x--x	0111	All classes can execute

### All together now

Let's use the examples from the symbolic notation section and show how it'd convert to octal notation

### CHMOD commands

---

Now that we have a better understanding of permissions and what all of these letters and numbers mean, let's take

Permission (symbolic notation)	CHMOD command	Description
-rwxrwxrwx	chmod 0777 filename; chmod -R 0777 dir	All classes can read/write/execute
-rwxr--r--	chmod 0744 filename; chmod -R 0744 dir	<b>user</b> can read/write/execute; all others can read
-rw-r--r--	chmod 0644 filename; chmod -R 0644 dir	<b>user</b> class can read/write; all others can read
-rw-rw-rw-	chmod 0666 filename' chmod -R 0666 dir	All classes can read/write

a look at how we can use the chmod command in our terminal to change permissions to anything we'd like!

These are just some examples. Using your new-found knowledge, you can set any permissions you'd like! Just be careful and make sure you don't break your system.

# Red Hat OpenShift Container Platform

## Cheat Sheet



### Table of Contents

1. What is OpenShift?.....	1	5. Simple routing overview .....	4
2. Cheat sheet guide.....	1	6. Examples.....	5
3. Command overview.....	2	7. Additional operations.....	10
4. Simple build and deploy overview.....	4	8. About the author.....	11

### What is OpenShift?

From OpenShift.com:

OpenShift is Red Hat's Platform-as-a-Service (PaaS) that allows developers to quickly develop, host, and scale applications in a cloud environment.

Openshift makes use of the Kubernetes upstream project to provide a secure, robust, and extendable manner for orchestrating applications. Openshift works to further the access management and build/deploy services that are provided in the upstream Kubernetes project. Development teams are empowered to own and maintain their applications through production environments, while operations teams can provide the guide rails for developers to have that application ownership in a multi-tenant environment.

### Cheatsheet Guide

This guide is mostly focused on the developer experience, however several administrator tasks are detailed below. A high-level listing of operations from command line interface is provided, followed by a walkthrough of an example application build, deploy, and management. The command list is not exhaustive, but does cover the majority of operations a developer will need to understand to manage an application's lifecycle.

# Command Overview

## Login/User management

`oc login` authenticate to an openshift cluster

`oc logout` end the current session

`oc whoami` show the current user context

## Project management

`oc project` show the current project context

`oc get projects` show all project current login has access to

`oc status` show overview of current project resources

`oc new-project` create a new project in Openshift and change to that context

## Resource management

`oc new-app` create a new application from source code, container image, or OpenShift template

`oc new-build` create a new build configuration from source code

`oc label` add/update/remove labels from an Openshift resource

`oc annotate` add/update/remove annotations from an Openshift resource

`oc create` create a new resource from filename or stdin

`oc get` retrieve a resource (use -o for additional output options)

`oc replace` replace an existing resource from filename or stdin

`oc delete` delete a resource

`oc edit` modify a resource from text editor

`oc describe` retrieve a resource with details

## Cluster management

`oc adm` administrative functions for an openshift cluster

`oc adm router|registry` install a router or registry

`oc adm policy` manage role/scc to user/group bindings, as well as additional policy administration

`oc adm diagnostics` run tests/validation against a cluster

`oc adm cordon/uncordon/drain` unschedule/schedule/drain a node

`oc adm groups` manage groups

`oc adm top` show usage statistics of resources

## Additional resource management

<code>oc patch</code>	Update fields for a resource with JSON or YAML segments
<code>oc extract</code>	get configmaps or secrets and save to disk
<code>oc set</code>	Modify miscellaneous application resources
<code>oc set probe</code>	Add a readiness/liveness probe on pod template/deployment configuration
<code>oc set volumes</code>	Manage volume types on a pod template/deployment configuration
<code>oc set build-hook</code>	Set a script/command to execute as part of the build process
<code>oc set build-secret</code>	set a secret to be included as part of the build process
<code>oc set env</code>	set environment variables on a pod template/deployment configuration/build configuration
<code>oc set image</code>	update the image for deployment configurations/daemonsets
<code>oc set triggers</code>	set triggers for deployment configurations/build configurations

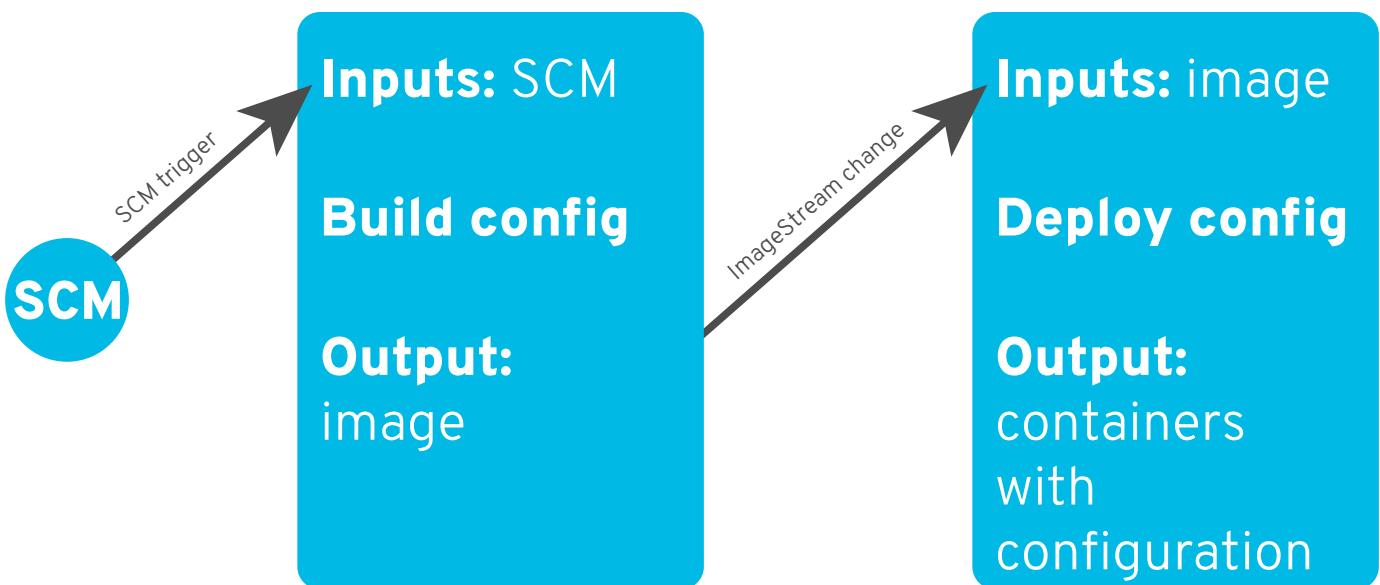
## Operational commands

<code>oc logs</code>	retrieve the logs for a resource (build configurations, deployment configurations, and pods)
<code>oc rsh</code>	remote shell into a container
<code>oc rsync</code>	copy files to or from a container
<code>oc exec</code>	execute a command in a container
<code>oc run</code>	create a deployment configuration from image
<code>oc idle</code>	scale resources to zero replicas

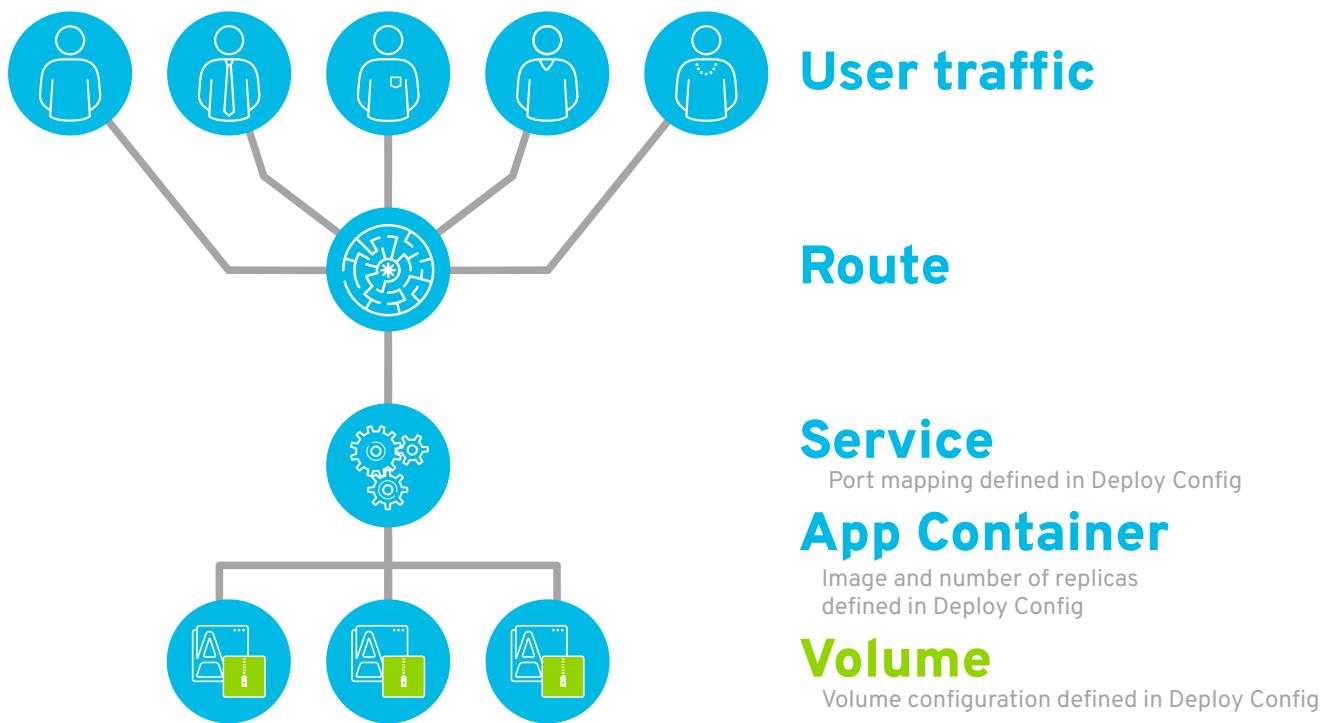
## Build / Deploy

<code>oc rollout</code>	manage deployments from deployment configuration
<code>oc rollout latest</code>	start a new deployment with the latest state
<code>oc rollout undo</code>	perform a rollback operation
<code>oc rollout history</code>	oc rollout history - View historical information for a deployment configuration
<code>oc rollout status</code>	watch the status of a rollout until complete
<code>oc tag</code>	tag existing images into image streams
<code>oc start-build</code>	start a new build from a build configuration
<code>oc cancel-build</code>	cancel a build in progress
<code>oc import-image</code>	pull in images and tags from an external Docker registry
<code>oc scale</code>	change the number of pod replicas for a deployment

## Simple Build/Deploy Overview



## Simple Routing Overview



## Examples

### Login

First, we can login to the cluster to interact with Openshift via CLI

```
$ oc login -u myuser https://openshift.example.com
Authentication required for https://openshift.example.com
Username: myuser
Password:
```

Note that leaving the -p option off of login prompts for password. Additionally we can verify our user context:

```
$ oc whoami
myuser
```

### Create Project

Let's list out our current available projects (those that we have at least view access for):

```
$ oc get projects
```

If this is our first login and no one has added us to any existing projects, there shouldn't be any projects listed. Let's create a project (allowed by self-provisioner role to all authenticated users, in the default Openshift policy installation).

```
$ oc new-project myproject --display-name='My Project' --description='cool project owned by myuser'
```

Now using project "myproject" on server "https://openshift.example.com:443".

To build a new example application on Ruby you can add applications to this project with the 'new-app' command. For example, try:

```
oc new-app centos/ruby-22-centos7~https://github.com/openshift/ruby-ex.git
```

If you want to view the specifics of the project definition, output the full spec to YA

```
$ oc get project myproject1 -o yaml
apiVersion: v1
kind: Project
metadata:
 annotations:
 openshift.io/description: A really cool project owned by myuser
 openshift.io/display-name: My Project
 openshift.io/requester: myuser
 openshift.io/sa.scc.mcs: s0:c51,c20
 openshift.io/sa.scc.supplemental-groups: 1000000000/10000
 openshift.io/sa.scc.uid-range: 1000000000/10000
 creationTimestamp: 2017-02-10T15:36:18Z
 labels:
 name: myproject
 resourceVersion: "32381158"
 selfLink: /oapi/v1/projects/myproject
 uid: aa94c906-efa6-11e6-af71-02a55ffb157d
 spec:
 finalizers:
 - openshift.io/origin
 - kubernetes
 status:
 phase: Active
```

## Add users to project

We can add additional users to our project by default, since self-provisioners get the "admin" role for any project they create:

```
$ oc adm policy add-role-to-user edit anotheruser
```

This allows anotheruser to edit resources within the project, but not manage policy

## Create app from code and image

```
$ oc new-app centos/ruby-22-centos7~https://github.com/openshift/ruby-ex.git
--> Found Docker image 06f0cdc (2 days old) from Docker Hub for "centos/ruby-22-
centos7"
Ruby 2.2

Platform for building and running Ruby 2.2 applicationsTags: builder, ruby, ruby22
* An image stream will be created as "ruby-22-centos7:latest" that will track the
source image
* A source build using source code from https://github.com/openshift/ruby-ex.git
will be created
* The resulting image will be pushed to image stream "ruby-ex:latest"
* Every time "ruby-22-centos7:latest" changes a new build will be triggered
* This image will be deployed in deployment config "ruby-ex"
* Port 8080/tcp will be load balanced by service "ruby-ex"
* Other containers can access this service through the hostname "ruby-ex"
--> Creating resources with label app=ruby-ex ...
imagestream "ruby-22-centos7" created
imagestream "ruby-ex" created
buildconfig "ruby-ex" created
deploymentconfig "ruby-ex" created
service "ruby-ex" created
--> Success
Build scheduled, use 'oc logs -f bc/ruby-ex' to track its progress.
Run 'oc status' to view your app.
```

The new-app command handles the majority of resource creation via template. Notice that deploymentconfig/buildconfig/service/imagestream were all set up.

## Get resources

We can view the resources that were created as part of the new-app command, as well as the build/deploy resources that were created automatically. Notice that the new-app automatically started a new build of our code, and the deployment config watches successful builds to know when to next rollout/deploy. A good place to start with viewing application status is checking the pods in your project:

```
$ oc get pods
NAME READY STATUS RESTARTS AGE
ruby-ex-1-a7y56 1/1 Running 0 24m
ruby-ex-1-build 0/1 Completed 0 26m
```

This shows us the build pod completed successfully. Additionally we can see that there is one ready and running pod deployed with our application.

The status command shows us similar results:

```
$ oc status -v
In project My Project (myproject1) on server https://openshift.example.com:443
svc/ruby-ex - 172.30.36.21:8080
dc/ruby-ex deploys istag/ruby-ex:latest <-
bc/ruby-ex source builds https://github.com/openshift/ruby-ex.git on istag/ruby-
22-centos7:latest
deployment #1 deployed 26 minutes ago - 1 pod
Warnings:
* dc/ruby-ex has no readiness probe to verify pods are ready to accept traffic or
ensure deployment is successful.
try: oc set probe dc/ruby-ex --readiness ...
View details with 'oc describe <resource>/<name>' or list everything with 'oc get
all'.
```

## Add a volume

If we want to attach a volume to our pods, the oc set volume command can be used:

```
$ oc set volume dc/ruby-ex --add --mount-path=/mnt/emptydir
info: Generated volume name: volume-7d1e8
deploymentconfigs/ruby-ex

$ oc get pods
NAME READY STATUS RESTARTS AGE
ruby-ex-1-a7y56 1/1 Running 0 2h
ruby-ex-1-build 0/1 Completed 0 2h
ruby-ex-2-deploy 0/1 ContainerCreating 0 5s
```

In this example, a simple emptyDir volume was attached, though the same command can be used for Persistent Volumes. Also notice that the deployment configuration has a ConfigChange trigger, so adding this volume automatically started a new deployment.

## Edit resource

Making a change to any Openshift resource is simple. Let's change the /mnt/emptydir mountpath above to /mnt/appdata:

```
$ oc edit dc ruby-ex
Please edit the object below. Lines beginning with a '#' will be ignored,
and an empty file will abort the edit. If an error occurs while saving this file
will be reopened with the relevant failures.
#
...
volumeMounts:
- mountPath:/mnt/emptydir /mnt/appdata
 name: volume-7d1e8
...
```

Saving the file in your text editor will update the resource, or report errors if validation did not succeed. Note that this change on the deployment config kicks off another deployment for our app.

## Start build

If a new build from source is desired:

```
$ oc start-build ruby-ex
build "ruby-ex-2" started
```

## Watch build

The build logs can be watched with the oc logs command (including -f option for follow):

```
$ oc logs -f bc/ruby-ex
Cloning "https://github.com/openshift/ruby-ex.git" ...
Commit: 855ab2de53ff897a19e1055f7554c64d19e02c50 (Merge pull request #6 from aj07/typo)
Author: Ionut Palade <PI-Victor@users.noreply.github.com>
Date: Mon Dec 12 14:37:32 2016 +0100
---> Installing application source ...
---> Building your Ruby application from source ...
---> Running 'bundle install --deployment --without development:test' ...
Fetching gem metadata from https://rubygems.org/.....
Installing puma 3.4.0
Installing rack 1.6.4
Using bundler 1.7.8
Your bundle is complete!
Gems in the groups development and test were not installed.
It was installed into ./bundle
---> Cleaning up unused ruby gems ...

Pushing image 172.30.114.236:5000/myproject/ruby-ex:latest ...
Pushed 7/9 layers, 78% complete
Pushed 8/9 layers, 89% complete
Pushed 9/9 layers, 100% complete
Push successful
```

## Start Deploy

Most configuration or image changes will automatically start a new deploy by default, but new deployments can be started manually as well:

```
$ oc rollout latest ruby-ex
deploymentconfig "ruby-ex" rolled out
```

## Watch Deploy

The overall deployment status can be watched via oc logs command:

```
$ oc logs -f dc/ruby-ex
--> Scaling up ruby-ex-5 from 0 to 1, scaling down ruby-ex-4 from 1 to 0 (keep 1
pods available, don't exceed 2 pods)
Scaling ruby-ex-5 up to 1
Scaling ruby-ex-4 down to 0
--> Success
```

Additionally container logs can be viewed with oc logs:

```
$ oc logs ruby-ex-5-kgzvd
[1] Puma starting in cluster mode...
[1] * Version 3.4.0 (ruby 2.2.2-p95), codename: Owl Bowl Brawl
[1] * Min threads: 0, max threads: 16
[1] * Environment: production
[1] * Process workers: 8
[1] * Phased restart available
[1] * Listening on tcp://0.0.0.0:8080
[1] Use Ctrl-C to stop
[1] - Worker 2 (pid: 29) booted, phase: 0
[1] - Worker 1 (pid: 25) booted, phase: 0
[1] - Worker 5 (pid: 41) booted, phase: 0
[1] - Worker 3 (pid: 33) booted, phase: 0
[1] - Worker 0 (pid: 21) booted, phase: 0
[1] - Worker 4 (pid: 37) booted, phase: 0
[1] - Worker 6 (pid: 45) booted, phase: 0
[1] - Worker 7 (pid: 60) booted, phase: 0
```

## Remote shell

Interacting directly with the container is simple with oc rsh:

```
$ oc rsh ruby-ex-5-kgzvd
sh-4.2$ ls
Gemfile Gemfile.lock README.md bundle config.ru
```

## Create route

```
$ oc expose service ruby-ex
route "ruby-ex" exposed
```

With no other options defined this will create a route for your application using the default route naming (ex: \$appname-\$projectname.openshift.example.com)

## Idle app

We're done testing our application, so we can idle the service in order to save resources. This interacts with a Kubernetes service to set the pod replicas to 0, and when the service is next accessed will automatically boot up the pods again:

```
$ oc idle ruby-ex
Marked service myproject1/ruby-ex to unidle resource DeploymentConfig myproject1/ruby-ex (unidle to 1 replicas)
Idled DeploymentConfig myproject1/ruby-ex
```

## Delete app

If we're completely done with our application, we can delete resources within the project (or the project itself) to clean up:

```
$ oc delete services -l app=ruby-ex
service "ruby-ex" deleted

$ oc delete all -l app=ruby-ex
buildconfig "ruby-ex" deleted
imagestream "ruby-22-centos7" deleted
imagestream "ruby-ex" deleted
deploymentconfig "ruby-ex" deleted

$ oc delete project myproject
project "myproject" deleted
```

## About the author



**BEN PRITCHETT** is a Systems Engineer within Red Hat's IT department, deploying Red Hat's internal offerings of Platform-as-a-Service technologies. His focus is to bridge the gap between development and operations teams using automation and training.

 <http://github.com/bjpritch>

# Bash Shell Scripting

Bash is a version of the classic Unix shell with many enhancements. Bash is the default shell installed on GNU/Linux distributions and many other Unix-style systems. This cheat sheet covers some useful concepts in Bash scripting.

If a command in the examples produces output, the output is shown on the same line, separated from the command by a hash or pound sign (#).

## BASH SCRIPT HEADER

```
#!/usr/bin/env bash
echo "Hello World"
```

## VARIABLES & STRINGS

```
#!/usr/bin/env bash
MSG="Hello World"
echo "$MSG Alex" # Hello World Alex
echo '$MSG Alex' # $MSG Alex
```

## String Manipulation

```
MSG="hello world"
Replace
echo ${MSG/w/W} # hello World
echo ${MSG//[a-zA-Z]/X} # XXXXX XXXXX
Substring
echo ${MSG:0:5} # hello
echo ${MSG%world} # hello
echo ${MSG#hello} # world
Uppercase
echo ${MSG^} # Hello world
echo ${MSG^^} # HELLO WORLD
MSG="HELLO WORLD"
echo ${MSG,,} # hELLO WORLD
echo ${MSG,,} # hello world
Alternative
echo ${MSG:-val} # HELLO WORLD
echo ${FOO:-val} # val
```

## FUNCTIONS

```
helloworld() {
 echo "Number of arguments $#" # 2
 echo "Hello World $1 from $2" # Hello World Alex from Bash
}
helloworld "Alex" "Bash"
```

## Returning Values

```
helloworld () {
 return 46
}
helloworld
echo $? # 46
```

Bash can return only a status code. To return a string, use command substitution:

```
helloworld() {
 echo 'My return string!'
```

```
}

msg=$(helloworld)
echo $msg
```

## COLLECTIONS

### Arrays

```
names=('Alex' 'Ada' 'Alexandra')
names+=('Soto') # Appends element
unset names[3] # Removes element

echo ${names[0]} # Alex
echo ${names[@]} # Alex Ada Alexandra
echo ${#names[@]} # 3
```

### Maps

```
declare -A score
score[alex]=1
score[edson]=2
score[sebi]=3
unset score[alex] # Delete alex entry

echo ${!score[@]} # alex edson sebi
echo ${score[@]} # 2 1 3
echo ${#score[@]} # 3
```

## CONDITIONALS

```
if [[$a -gt 4]]; then
 echo "$a is greater than 4"
elif [[$a -lt 4]]; then
 echo "$a less than 4"
else
 echo "$a is equal 4"
fi
```

## Numeric Conditions

```
[[NUM -eq NUM]] Equal
[[NUM -ne NUM]] Not equal
[[NUM -lt NUM]] Less than
[[NUM -le NUM a]] Less than or equal to a
[[NUM -gt NUM]] [[NUM -ge NUM]] Greater than or equal to
```

Greater than

## String Conditions

```
[[STRING
== STRING Equal
]]
[[STRING != STRING]] Not Equal
[[-z STRING]] Empty string
[[-n STRING]] Not empty string
[[STRING =~ STRING]] Regular expression match
```

## File Conditions

```
[[-f FILE]] Is a file
[[-d FILE]] Is a directory
[[-e FILE]] Exists
[[-r -w -x FILE]] Is readable, Writable, executable
[[-h FILE]] Is symbolic link
Boolean conditions:
a|[[! EXPR]] | Not
a|[[BOOL && BOOL]] |And
a|[[BOOL || BOOL]] |OR
```

## LOOPS

```
for ((i = 0 ; i < 10 ; i++)); do
 echo "Hello World $i"
done
```

## Range

```
for i in {1..5}; do
 echo "Hello World $i"
done
```

## Collections

Print all elements from a plain array:

```
for i in "${names[@]}"; do
 echo "Hello $i"
done
```

Print keys and values of all elements from a key/value array:

```
for key in "${!score[@]}"; do
 echo $key
done

for val in "${score[@]}"; do
 echo $val
done
```

## Files

```
for i in /tmp/*.txt; do
 echo $i
```

done

```
cat /tmp/hello.txt | while read line; do
 echo $line
done
```

## While

```
x=1;
while [$x -le 5]; do
 echo "Hello World"
done
```

## EXECUTING COMMANDS

Execute a command and check the exit status:  
`cat /tmp/hello.txt`

```
if [$? -eq 0]
then
 echo "OK"
else
 echo "KO"
fi
```

To get the output of a command, surround the call with "\`" character:  
`lines=(`cat "/tmp/hello.txt"`)`  
`lines=$(cat "/tmp/hello.txt")`

## USEFUL SNIPPETS

### Getting the Script Directory

```
DIR="${0%/*}"
```

### Reading CLI Arguments:

```
echo "$1 $2"
#####
execute.sh "Hello" "Alex"
Hello Alex
```

## Print Output

```
printf "\n\n##### Deploying #####\n"
```

## Read Input

```
echo -n "Enter name: "
read ans
echo $ans
```

## Create File with Content

```
echo "
apiVersion: apiserver.config.k8s.io/v1
kind: EncryptionConfiguration
resources:
- resources:
 - secrets
 providers:
```

```
- aescbc:
 keys:
 - name: key1
 secret: b6sjdRWAPhtacXo8mO1cfgVYWXzwuls3T3NQOo4TBhk=
 identity: {}
"
| tee /var/lib/minikube/certs/encryptionconfig.yaml
```

```
ps -ef | grep execute.sh
501 4286 641 0 11:17AM ttys007 0:00.00 /bin/bash ./execute.sh
501 4287 4286 0 11:17AM ttys007 0:07.67 /bin/bash ./execute.sh
```

Two processes are started. The first one (4286) as parent of the second.

## Subshell

A shell script can launch subshells. These subshells let the script do parallel processing, in effect executing multiple subtasks simultaneously.

```
(
Inside parentheses, and therefore a subshell . . .
while [1] # Endless loop.
do
 echo "Subshell"
done
)
```

Run the following command in a new terminal:

# YUM COMMAND CHEAT SHEET

for Red Hat Enterprise Linux

## YUM QUERIES

### SUBCOMMAND DESCRIPTIONS AND TASKS

**help** Display yum commands and options  
**yum help** Show yum subcommands and options

### Individual packages

**list** List package names from repositories  
**yum list available** List all available packages  
**yum list installed** List all installed packages  
**yum list all** List installed and available packages  
**yum list kernel** List installed and available kernel packages

**info** Display information about a package  
**yum info vsftpd** List info about vsftpd package

**deplist** Display dependencies for a package  
**yum deplist nfs-utils** List dependencies and packages providing them

**provides** Find packages that provide the queried file  
**yum provides “\*bin/top”** Show package that contains top command  
**yum provides “\*/README.top”** Show package containing README.top file

**search** Search package names and descriptions for a term  
**yum search samba** Find packages with samba in name or description

**updateinfo** Get information about available package updates  
**yum updateinfo security** Get info on available security updates

### Groups of packages

**grouplist** List names of installed and available package groups  
**groupinfo** Display description and contents of a package group  
**yum groupinfo “Web Server”** See packages in Web Server group

**check-update** Query repositories for available package updates

## MANAGE YUM REPOSITORIES

### SUBCOMMAND DESCRIPTIONS AND TASKS

**repolist** Display enabled software repositories

**repoinfo** Display information about enabled yum repositories \*  
**yum repoinfo rhel-7-server-rpms** See info on rhel-7-server-rpms repo

**repo-pkgs** Work with packages in a particular repository \*  
**yum repo-pkgs my-rpms list** List packages from my-rpms repo  
**yum repo-pkgs my-rpms install** Install all packages from my-rpms repo  
**yum repo-pkgs my-rpms remove** Remove all packages from my-rpms repo

**makecache** Download yum repository data to cache

## TROUBLESHOOT AND MAINTAIN YUM

### SUBCOMMAND DESCRIPTIONS AND TASKS

**check** Check the local RPM database for problems (runs for a long time)

**history** View and use yum transactions  
**yum history list** List all yum install, update and erase actions  
**yum history info 3** Show details of yum transaction 3  
**yum history undo 3** Undo the yum action from transaction 3  
**yum history redo 3** Redo the undone yum action from transaction 3

**clean** Clear out cached package data  
**yum clean packages** Delete packages saved in cache  
**yum clean all** Clean out all packages and meta data from cache

**fssnapshot** List LVM snapshots (helps roll back after package updates)  
**fs** Act on filesystem (prevent doc or language file install on minimal systems)  
**yum fs filters** List enabled filesystem filters  
**yum fs documentation** Filters all docs from being installed (careful!)

## INSTALL, REMOVE AND UPGRADE PACKAGES WITH YUM

### SUBCOMMAND DESCRIPTIONS AND TASKS

**install** Install a package from a repository to your system  
**yum install vsftpd** Install the vsftpd package

**update** Update one or all packages on your system  
**yum update** Update all packages with available updates  
**yum update httpd** Update the httpd package (if available)  
**yum update --security** Apply security-related package updates

**update-to** Update one or all packages to a particular version

**upgrade** Update packages taking obsoletes into account

**localinstall** Install a package from a local file, http, or ftp  
**yum localinstall abc-1-1.i686.rpm** Install abc package from local directory  
**yum localinstall http://myrepo/abc-1-1.i686.rpm** Install abc from FTP site

**downgrade** Downgrade a package to an earlier version  
**yum downgrade abc** Downgrade the abc package to an earlier version

**reinstall** Reinstall the current version of a package  
**yum reinstall util-linux** Reinstall util-linux (to replace any deleted files)

**swap** Remove one package and install another  
**yum swap ftp lftp** Remove ftp package and install lftp package

**erase** Erase a package (and possibly dependencies) from your system  
**yum remove vsftpd** Remove the vsftpd package and dependencies

**remove** Same as erase

**autoremove** Same as erase, plus removes additional unneeded packages \*  
**yum autoremove httpd** Remove httpd and other unneeded packages

**groupinstall** Install all packages in the selected group  
**yum groupinstall “Web server”** Install Web Server packages

## MANAGE LANGUAGE PACKAGES WITH YUM

SUBCOMMAND	DESCRIPTIONS AND TASKS
<b>langavailable</b>	List all available languages *
<b>langinfo</b>	List packages available for a language * <code>yum langinfo es</code> List packages associated with Spanish language
<b>langinstall</b>	Install packages associated with a particular language * <code>yum langinstall es</code> Install packages associated with Spanish language
<b>langlist</b>	List languages that are installed *
<b>langremove</b>	Remove installed language packs for a language * <code>yum langremove es</code> Remove packages associated with Spanish language

## POPULAR OPTIONS FOR DIFFERENT YUM COMMANDS \*\*

OPTION	DESCRIPTION
<b>-y</b>	Assume yes if prompted
<b>--assumeno</b>	Assume no if prompted
<b>-q</b>	Produce no output
<b>-v</b>	Produce extra debugging output
<b>--nopugins</b>	Run command without loading any yum plugins
<b>--disableplugin=</b>	Disable a particular plugin for single command <code>yum --disableplugin=langpacks info vsftpd</code>
<b>--enableplugin=</b>	Enable a plugin that is installed, but currently disabled <code>yum --enableplugin=ps ps</code> Show packages tied to running processes
<b>--enablerepo=</b>	Enable currently disabled repo for a single command (wildcards okay) <code>yum install docker \--enablerepo=rhel-7-server-extras-rpm</code>
<b>--disablerepo=</b>	Disable currently enabled repo for a single command (wildcards okay) <code>yum list available --disablerepo=epel</code>
<b>--downloadonly</b>	Download to /var/cache/yum/arch/prod/repo/ packages/, but don't install <code>yum install --downloadonly vsftpd</code> Download vsftpd package to cache
<b>--filter-???</b>	Replace ??? with vendors, rpm-groups, arches, and others to filter output
<b>--changelog</b>	Display changelog information of package

## MORE YUM-RELATED COMMANDS (install the yum-utils package)

COMMAND	DESCRIPTION
<b>find-repos-of-install</b>	Find which repository a package comes from
<b>needs-restarting</b>	Find processes that have been updated and need to restart
<b>repoclosure</b>	Get unmet dependency list from repositories
<b>repoquery</b>	Query remote repos and local RPM database <code>repoquery --requires --resolve bash</code> Show dependent packages
<b>reposync</b>	Synchronize yum repositories to a local directory <code>reposync -r rhel-atomic-host-beta-rpms</code> Get packages from repo
<b>repotrack</b>	Download a package and all its dependencies
<b>show-installed</b>	List installed RPM packages and statistics
<b>verifytree</b>	Check the local yum repository for consistency
<b>yum-complete-transaction</b>	Try to complete yum transactions that didn't finish
<b>yumdb</b>	Check or change the yum database
<b>yumdownloader</b>	Download a package from a repo to current directory

Type `man yum` for further details on all yum subcommands and options

\* New options for RHEL 7

\*\* Some options need yum plugins. Type `yum list "yum-plugin*"` to see available plugins.

# RED HAT ENTERPRISE LINUX 5, 6, AND 7

## Common administrative commands

	RHEL5	RHEL6	RHEL7	
<b>SYSTEM BASICS</b>				
View subscription information	/etc/sysconfig/rhn/systemid	/etc/sysconfig/rhn/systemid subscription-manager identity	subscription-manager identity	
Configure subscription	rhn_register	rhn_register rhnreg_ks subscription-manager	subscription-manager <sup>1</sup> rhn_register <sup>2</sup>	
View system profile	sosreport dmidecode hwbrowser		sosreport dmidecode lshw	
View RHEL version information			/etc/redhat-release	
	1 subscription-manager is used for Satellite 6, Satellite 5.6 with SAM and newer, and Red Hat's CDN. 2 RHN tools are deprecated on Red Hat Enterprise Linux 7. rhn_register should be used for Satellite server 5.6 and newer only. For details, see: <a href="#">Satellite 5.6 unable to register RHEL 7 client system due to rhn-setup package not included in Minimal Installation</a>			
<b>BASIC CONFIGURATION</b>				
Graphical configuration tools		system-config-*	gnome-control-center	
Configure network		system-config-network	nmcli nmui nm-connection-editor	
Configure system language		system-config-language	localectl	
Configure time and date		system-config-date	timedatectl date	
Synchronize time and date	ntpdate /etc/ntp.conf		timedatectl /etc/chrony.conf	
Configure keyboard		system-config-keyboard	localectl	
Text-based configuration tools		system-config-* tui		
Configure printer		system-config-printer		
Configure samba		smbclient /etc/samba/smb.conf smbpasswd		
Configure SSH		/etc/ssh/ssh_config /etc/ssh/sshd_config -/.ssh/config ssh-keygen		
<b>JOBs AND SERVICES</b>				
Configure logging	/etc/syslog.conf	/etc/rsyslog.conf	/etc/rsyslog.conf /etc/rsyslog.d/*.*.conf /var/log/journal systemd-journal.service	
List all services		chkconfig --list ls /etc/init.d/	systemctl -at service ls /etc/systemd/system/*.service ls /usr/lib/systemd/system/*.service	
List running services		service --status-all	systemctl -t service --state=active	
Start/stop service		service name start service name stop	systemctl start name.service systemctl stop name.service	
Enable/disable service		chkconfig name on chkconfig name off	systemctl enable name.service systemctl disable name.service	
View service status		service name status	systemctl status name.service	
Check if service is enabled		chkconfig name	systemctl is-enabled name	
Create new service file or modify configuration		chkconfig --add	systemctl daemon-reload	
View run level/target		runlevel who -r	systemctl get-default who -r	
Change run level/target		/etc/inittab init run_level	systemctl isolate name.target systemctl set-default	
View logs		/var/log	/var/log journalctl	
Configure system audit		add audit=1 to kernel cmdline audictl /etc/audit/auditd.conf /etc/audit/audit.rules authconfig /etc/pam.d/system-auth pam_tty_audit kernel module		
View audit output		aureport /var/log/faillog		
Schedule tasks		cron at		
Configure batch tasks		batch		
Find file by name		locate		
Find file by characteristic		find		
Create archive		tar cpio zip gzip bzip2		
<b>KERNEL, BOOT, AND HARDWARE</b>				
Single user/rescue mode	append 1 or s or init=/bin/bash to kernel cmdline		append rd.break or init=/bin/bash to kernel cmdline	
Shut down system	shutdown		systemctl shutdown	
Power off system	poweroff		systemctl poweroff	
Halt system	halt		systemctl halt	
Reboot system	reboot		systemctl reboot	
Configure default run level/target	/etc/inittab		systemctl set-default	
Configure GRUB bootloader	/boot/grub/grub.conf		/etc/default/grub grub2-mkconfig grub-set-default	
View hardware configured	hwbrowser	lshw		
Configure kernel module		modprobe		
Configure hardware device		udev		
View kernel parameters		sysctl -a cat /proc/cmdline		
Load kernel module		modprobe		
Remove kernel module		modprobe -r		
View kernel version		rpm -q kernel uname -r		
<b>SOFTWARE MANAGEMENT</b>				
Install software		yum install yum groupinstall	yum install yum group install	
View software info		yum info yum groupinfo	yum info yum group info	
Update software		yum update		
Upgrade software		yum upgrade		
Configure software repository		/etc/yum.repos.d/*.repo		
Find file in package		rpm -qf filename		
View software version		rpm -q packagename		
View installed software		rpm -qa		
<b>SECURITY AND IDENTITY</b>				
Configure system security		/etc/selinux/config chcon restorecon semanage setsebool system-config-selinux		
Report on system security		sealert		
LDAP, SSSD, Kerberos		authconfig authconfig-tui authconfig-gtk		
Network users		getent		
<b>FILE SYSTEMS, VOLUMES, AND DISKS</b>				
Default file system		ext3	ext4	xfs
Defragment disk space		copy data to new file system fsck (look for 'non-contiguous inodes')	copy data to new file system fsck (look for 'non-contiguous inodes') xfs_fsr	
Create/modify disk partitions		fdisk parted	fdisk gdisk parted ssm create	
Format disk partition		mkfs.filesystem_type (ext4, xfs) mkswap	mkfs.filesystem_type (ext4, xfs) mkswap ssm create	
Mount storage		mount /etc/fstab	mount /etc/fstab ssm mount	
Create physical volume		pvcreate	pvcreate ssm create (if backend is lvm)	
Create volume group		vgcreate	vgcreate ssm create (if backend is lvm)	
Create logical volume		lvcreate	lvcreate ssm create (if backend is lvm)	
Enlarge volumes formatted with default file system		vgextend lvextend resize2fs	vgextend lvextend resize2fs	xfs cannot currently be shrunk; copy desired data to a smaller file system.
Shrink volumes formatted with default file system		resize2fs lvreduce vgreduce		
Check/repair file system		fsck	fsck ssm check	
Configure NFS share		/etc/exports service nfs reload	/etc/exports systemctl reload nfs.service	
Mount and activate swap			/etc/fstab swapon -a	
Automatically mount at boot			/etc/fstab	
View free disk space			df	
View logical volume info			lvdisplay lvs vgdisplay vgs pvdisk pvs	
View NFS share			showmount -e mount	
Automatically mount after boot			/etc/auto.master.d/*. autofs /etc/auto.*	
Change file permissions			chmod chown chgrp umask	
Change access control list			setfacl	
<b>NETWORKING</b>				
Configure firewall		iptables and ip6tables /etc/sysconfig/iptables	iptables and ip6tables /etc/sysconfig/iptables system-config-firewall	firewall-cmd firewall-config
Configure DHCP client		/etc/dhcpd.conf /etc/dhcp6c.conf		dhcpd /etc/dhcp/dhcpd.conf /etc/sysconfig/dhcpd
Configure name resolution			/etc/hosts /etc/resolv.conf	/etc/hosts /etc/resolv.conf nmcli con mod
Configure hostname			/etc/sysconfig/network	hostnamectl /etc/hostname nmui
View network interface info			ip addr ifconfig brctl	ip addr nmcli dev show teamdctl brctl bridge
Configure network interface			/etc/sysconfig/network-scripts/ifcfg-*	/etc/sysconfig/network-scripts/ifcfg-* nmcli con [add mod edit] nmui nm-connection-editor
View ports/sockets			ss lsof netstat	ss lsof
View routes			ip route	
Configure routes			/etc/sysconfig/network system-config-network	
<b>RESOURCE MANAGEMENT</b>				
View system usage		top ps sar iostat netstat vmstat mpstat numastat	top ps sar iostat netstat ss vmstat mpstat numastat tuna	top ps sar iostat ss vmstat mpstat numastat tuna
View disk usage		df		df iostat
Trace system calls				strace
Trace library calls				ltrace
Change process priority				nice renice
Change process run location				taskset
Kill a process				kill pkill killall

# Linux Commands Cheat Sheet

Easy to use Linux shortcuts  
for developers.



---

**ssh [ip or hostname]**  
"vagrant ssh" in the same directory as the Vagrantfile to shell into the box/machine (assumes you have successfully "vagrant up")

Secure shell, an encrypted network protocol allowing for remote login and command execution  
On Windows: PuTTY and WinSCP  
An "ssh.exe" is also available via Cygwin as well as with a Git installation.

---

**pwd**

Print Working Directory  
Displays the full path name

---

**whoami**

Displays your logged in user id

---

**cd /**  
**cd target**  
**cd ~**

Change directory to the root of the filesystem  
Change directory to "target" directory  
Change directory to your home directory

---

**ls**

Directory listing  
**ls -l**  
Long listing, displays file ownership  
**ls -la**  
Displays hidden files/directories

---

```
[vagrant@rhel-cdk /]$ ls
bin boot dev etc home lib lib64 lost+found media mnt opt pro
[vagrant@rhel-cdk /]$ ls -l
total 62
lrwxrwxrwx. 1 root root 7 Mar 8 20:36 bin -> usr/bin
dr-xr-xr-x. 4 root root 1024 Mar 12 19:26 boot
drwxr-xr-x. 18 root root 3100 Mar 12 19:49 dev
drwxr-xr-x. 85 root root 4096 Mar 12 19:31 etc
drwxr-xr-x. 3 root root 4096 Mar 8 20:54 home
lrwxrwxrwx. 1 root root 7 Mar 8 20:36 lib -> usr/lib
lrwxrwxrwx. 1 root root 9 Mar 8 20:36 lib64 -> usr/lib64
drwx-----. 2 root root 16384 Mar 8 20:34 lost+found
drwxr-xr-x. 2 root root 4096 May 25 2015 media
drwxr-xr-x. 2 root root 4096 May 25 2015 mnt
drwxr-xr-x. 3 root root 4096 Mar 8 20:50 opt
dr-xr-xr-x. 166 root root 0 Mar 12 19:26 proc
dr-xr-x---. 3 root root 4096 Mar 12 19:30 root
drwxr-xr-x. 31 root root 1000 Mar 12 19:31 run
lrwxrwxrwx. 1 root root 8 Mar 8 20:36 sbin -> usr/sbin
drwxr-xr-x. 2 root root 4096 May 25 2015 srv
dr-xr-xr-x. 13 root root 0 Mar 12 19:26 sys
drwxrwxrwt. 7 root root 4096 Mar 12 20:31 tmp
drwxr-xr-x. 13 root root 4096 Mar 8 20:36 usr
drwxr-xr-x. 3 vagrant vagrant 4096 Mar 12 19:25 vagrant
drwxr-xr-x. 19 root root 4096 Mar 12 19:26 var
[vagrant@rhel-cdk /]$ □
```

---

**clear**

Clear the terminal screen

---

<b>cat file.txt</b>	Displays the contents of file.txt to standard out
<b>cat /etc/system-release</b>	Displays the contents of the system-release file - what version of RHEL, Centos or Fedora are you running?

```
[vagrant@rhel-cdk etc]$ cat /etc/system-release
Red Hat Enterprise Linux Server release 7.2 (Maipo)
[vagrant@rhel-cdk etc]$]
```

---

<b>cat longfile.txt   more</b>	Displays the contents of the file with forward paging
	<pre>Count is 0 Count is 1 Count is 2 Count is 3 Count is 4 Count is 5 Count is 6 Count is 7 Count is 8 Count is 9 Count is 10 Count is 11 Count is 12 Count is 13 Count is 14 Count is 15 Count is 16 Count is 17 Count is 18 Count is 19 Count is 20 Count is 21 Count is 22 Count is 23 Count is 24 Count is 25 Count is 26 Count is 27 --More--]</pre>

---

<b>less longfile.txt</b>	Scroll forward: Ctrl-f Scroll backward: Ctrl-b End of file: G Quit less: q
--------------------------	-------------------------------------------------------------------------------------

---

<b>man cat</b>	Man pages, the user manual. In this case, it will describe the cat command
	<pre>CAT(1)  NAME     cat - concatenate files and print on the standard output  SYNOPSIS     cat [OPTION]... [FILE]...  DESCRIPTION     Concatenate FILE(s), or standard input, to standard output.      -A, --show-all         equivalent to -vET      -b, --number-nonblank         number nonempty output lines, overrides -n      -e    equivalent to -vE      -E, --show-ends         display \$ at end of each line      -n, --number         number all output lines      -s, --squeeze-blank         suppress repeated empty output lines      -t    equivalent to -vT      -T, --show-tabs         display TAB characters as ^I  Manual page cat(1) line 1 (press h for help or q to quit)]</pre>

---

<code>cp source_file.js target_file.js</code>	Copies a specific file
<code>cp -r ~/source_dir ~/target_dir</code>	Copies all files and sub-dirs

---

<code>mkdir my_directory</code>	Create the directory "my_directory"
---------------------------------	-------------------------------------

```
[vagrant@rhel-cdk ~]$ mkdir my_directory
[vagrant@rhel-cdk ~]$ ls -la
total 32
drwx----- 5 vagrant vagrant 4096 Mar 12 21:39 .
drwxr-xr-x 3 root root 4096 Mar 8 20:54 ..
-rw-r--r-- 1 vagrant vagrant 18 Jul 8 2015 .bash_logout
-rw-r--r-- 1 vagrant vagrant 193 Jul 8 2015 .bash_profile
-rw-r--r-- 1 vagrant vagrant 231 Jul 8 2015 .bashrc
drwxr-xr-x 2 vagrant vagrant 4096 Mar 12 19:26 .docker
drwxrwxr-x 2 vagrant vagrant 4096 Mar 12 21:39 my_directory
drwx----- 2 vagrant vagrant 4096 Mar 12 19:26 .ssh
[vagrant@rhel-cdk ~]$
```

---

<code>rm myfile.js</code>	Removes a specific file
<code>rm -rf my_directory</code>	Removes a directory, recursively

---

<code>mv [source_file] [target_file]</code>	Move file or directory
---------------------------------------------	------------------------

---

<code>ps -ef</code>	Displays information about a selection of the active processes
---------------------	----------------------------------------------------------------

---

```
[vagrant@rhel-cdk etc]$ ps -ef
UID PID PPID C STIME TTY TIME CMD
root 1 0 0 19:26 ? 00:00:02 /usr/lib/systemd/systemd
root 2 0 0 19:26 ? 00:00:00 [kthreadd]
root 3 2 0 19:26 ? 00:00:01 [ksoftirqd/0]
root 5 2 0 19:26 ? 00:00:05 [kworker/0:0H]
root 7 2 0 19:26 ? 00:00:00 [migration/0]
root 8 2 0 19:26 ? 00:00:00 [rcu_bh]
root 9 2 0 19:26 ? 00:00:00 [rcuob/0]
root 10 2 0 19:26 ? 00:00:00 [rcuob/1]
root 11 2 0 19:26 ? 00:00:19 [rcu_sched]
root 12 2 0 19:26 ? 00:00:14 [rcuos/0]
root 13 2 0 19:26 ? 00:00:16 [rcuos/1]
root 14 2 0 19:26 ? 00:00:00 [watchdog/0]
root 15 2 0 19:26 ? 00:00:00 [watchdog/1]
root 16 2 0 19:26 ? 00:00:00 [migration/1]
root 17 2 0 19:26 ? 00:00:00 [ksoftirqd/1]
root 19 2 0 19:26 ? 00:00:00 [kworker/1:0H]
root 20 2 0 19:26 ? 00:00:00 [khelper]
root 21 2 0 19:26 ? 00:00:00 [kdevtmpfs]
root 22 2 0 19:26 ? 00:00:00 [netns]
root 23 2 0 19:26 ? 00:00:00 [perf]
root 24 2 0 19:26 ? 00:00:00 [writeback]
root 25 2 0 19:26 ? 00:00:00 [kintegrityd]
root 26 2 0 19:26 ? 00:00:00 [bioset]
root 27 2 0 19:26 ? 00:00:00 [kblockd]
root 28 2 0 19:26 ? 00:00:00 [md]
root 29 2 0 19:26 ? 00:00:02 [kworker/0:1]
root 34 2 0 19:26 ? 00:00:00 [khungtaskd]
root 35 2 0 19:26 ? 00:00:01 [kswapd0]
root 36 2 0 19:26 ? 00:00:00 [ksmd]
```

## ./runthisthing

Execute a program or shell script in your current working directory (pwd)

Executable items are have an "x" in their long listing (ls -la)

```
[vagrant@rhel-cdk bin]$ pwd
/usr/bin
[vagrant@rhel-cdk bin]$ ls -la | grep cp
-rwxr-xr-x. 1 root root 155136 Nov 25 10:55 cp
-rwxr-xr-x. 1 root root 141632 Jul 8 2015 cpio
-rwxr-xr-x. 1 root root 768592 Jul 15 2015 cpp
-rwxr-xr-x. 1 root root 67928 Jan 23 05:07 cpupower
-rwxr-xr-x. 1 root root 58352 Aug 21 2015 lscpu
-rwxr-xr-x. 1 root root 11400 Sep 15 05:51 rpm2cpio
-rwxr-xr-x. 1 root root 70360 Jan 13 10:22 scp
[vagrant@rhel-cdk bin]$]
```

```
[vagrant@rhel-cdk ~]$ cd ~
[vagrant@rhel-cdk ~]$ ls -la
total 32
drwx----- 4 vagrant vagrant 4096 Mar 12 21:04 .
drwxr-xr-x. 3 root root 4096 Mar 8 20:54 ..
-rw-r--r--. 1 vagrant vagrant 18 Jul 8 2015 .bash_logout
-rw-r--r--. 1 vagrant vagrant 193 Jul 8 2015 .bash_profile
-rw-r--r--. 1 vagrant vagrant 231 Jul 8 2015 .bashrc
drwxr-xr-x. 2 vagrant vagrant 4096 Mar 12 19:26 .docker
-rwxrwxr-x. 1 vagrant vagrant 107 Mar 12 21:04 runthisthing
drwx----- 2 vagrant vagrant 4096 Mar 12 19:26 .ssh
[vagrant@rhel-cdk ~]$]
```

## ./runthisthing &

Execute a program or shell script as a background task

## ps -ef | grep runthisthing

Find a particular process by name. The "|" is a pipe, redirects the output of the left-side command to the standard input of the right-side command

```
[vagrant@rhel-cdk ~]$./runthisthing &
[3] 22047
[vagrant@rhel-cdk ~]$ ps -ef | grep runthisthing
vagrant 22047 14771 99 21:11 pts/0 00:00:05 /bin/bash ./runthisthing
vagrant 22060 14771 0 21:11 pts/0 00:00:00 grep --color=auto runthisthing
[vagrant@rhel-cdk ~]$]
```

## kill -9 [pid]

```
[vagrant@rhel-cdk ~]$ ps -ef | grep runthisthing
vagrant 22047 14771 99 21:11 pts/0 00:00:21 /bin/bash ./runthisthing
vagrant 22081 14771 0 21:11 pts/0 00:00:00 grep --color=auto runthisthing
[vagrant@rhel-cdk ~]$ kill -9 22047
[vagrant@rhel-cdk ~]$ ps -ef | grep runthisthing
vagrant 22096 14771 0 21:11 pts/0 00:00:00 grep --color=auto runthisthing
[3] Killed ./runthisthing
[vagrant@rhel-cdk ~]$]
```

## ip -4 a

Shows the IPv4 address for all NICs

---

**top**

What is eating your CPU

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
22526	vagrant	20	0	113116	1184	1008	R	100.0	0.1	0:08.06	runthisthing
13368	root	20	0	1448088	233012	15992	S	10.0	22.9	12:36.47	openshift
13168	root	20	0	1377296	44816	11004	S	1.3	4.4	1:10.37	docker
453	root	20	0	43588	9560	6892	S	0.3	0.9	0:10.46	systemd-journal
1	root	20	0	43896	5532	3568	S	0.0	0.5	0:02.06	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.03	kthreadadd
3	root	20	0	0	0	0	R	0.0	0.0	0:01.45	ksoftirqd/0
5	root	0	-20	0	0	0	S	0.0	0.0	0:06.25	kworker/0:0H
7	root	rt	0	0	0	0	S	0.0	0.0	0:00.66	migration/0
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
9	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcuob/0
10	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcuob/1
11	root	20	0	0	0	0	S	0.0	0.0	0:24.92	rcu_sched
12	root	20	0	0	0	0	R	0.0	0.0	0:18.26	rcuos/0
13	root	20	0	0	0	0	S	0.0	0.0	0:21.08	rcuos/1
14	root	rt	0	0	0	0	S	0.0	0.0	0:00.19	watchdog/0
15	root	rt	0	0	0	0	S	0.0	0.0	0:00.20	watchdog/1
16	root	rt	0	0	0	0	S	0.0	0.0	0:00.72	migration/1
17	root	20	0	0	0	0	S	0.0	0.0	0:01.08	ksoftirqd/1
19	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/1:0H
20	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	khelper
21	root	20	0	0	0	0	S	0.0	0.0	0:00.02	kdevtmpfs
22	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	netns
23	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	perf
24	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	writeback
25	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kintegrityd

---

which [executable]

Where is the executable located

```
[vagrant@rhel-cdk ~]$ which docker
/usr/bin/docker
[vagrant@rhel-cdk ~]$ which oc
/usr/bin/oc
[vagrant@rhel-cdk ~]$ which top
/usr/bin/top
[vagrant@rhel-cdk ~]$]
```

---

```
echo "Stuff" > target_file.txt
echo "more" >> target_file.txt
```

single > redirects the output to the file  
"target\_file.txt"  
A double >> appends

```
[vagrant@rhel-cdk ~]$ cd ~
[vagrant@rhel-cdk ~]$ ls -la
total 32
drwx----- 4 vagrant vagrant 4096 Mar 12 21:29 .
drwxr-xr-x. 3 root root 4096 Mar 8 20:54 ..
-rw-r--r--. 1 vagrant vagrant 18 Jul 8 2015 .bash_logout
-rw-r--r--. 1 vagrant vagrant 193 Jul 8 2015 .bash_profile
-rw-r--r--. 1 vagrant vagrant 231 Jul 8 2015 .bashrc
drwxr-xr-x. 2 vagrant vagrant 4096 Mar 12 19:26 .docker
-rwxrwxr-x. 1 vagrant vagrant 114 Mar 12 21:11 runthisthing
drwx----- 2 vagrant vagrant 4096 Mar 12 19:26 .ssh
[vagrant@rhel-cdk ~]$ echo "stuff" > target_file.txt
[vagrant@rhel-cdk ~]$ cat target_file.txt
stuff
[vagrant@rhel-cdk ~]$ echo "more" >> target_file.txt
[vagrant@rhel-cdk ~]$ cat target_file.txt
stuff
more
[vagrant@rhel-cdk ~]$ ls -la
total 36
drwx----- 4 vagrant vagrant 4096 Mar 12 21:29 .
drwxr-xr-x. 3 root root 4096 Mar 8 20:54 ..
-rw-r--r--. 1 vagrant vagrant 18 Jul 8 2015 .bash_logout
-rw-r--r--. 1 vagrant vagrant 193 Jul 8 2015 .bash_profile
-rw-r--r--. 1 vagrant vagrant 231 Jul 8 2015 .bashrc
drwxr-xr-x. 2 vagrant vagrant 4096 Mar 12 19:26 .docker
-rwxrwxr-x. 1 vagrant vagrant 114 Mar 12 21:11 runthisthing
drwx----- 2 vagrant vagrant 4096 Mar 12 19:26 .ssh
-rw-rw-r--. 1 vagrant vagrant 11 Mar 12 21:29 target_file.txt
[vagrant@rhel-cdk ~]$]
```

---

**echo \$PATH**

Displays the \$PATH environment variable

```
[vagrant@rhel-cdk ~]$ echo $PATH
/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/vagrant/.local/bin:/home/vagrant/bin
```

---

**env**

Displays all ENV variables

---

**export PATH=\$PATH:/anotherdir**

Adds "anotherdir" to your PATH, just for your current session

---

**sudo find . -name [file]**

Find a file or directory by name

```
[vagrant@rhel-cdk /]$ cd /
[vagrant@rhel-cdk /]$ sudo find . -name vagrant
.var/spool/mail/vagrant
.vagrant
.vagrant/src/booker/vagrant
.etc/sudoers.d/vagrant
.home/vagrant
[vagrant@rhel-cdk /]$]
```

---

---

```
grep -i stuff `find . -name *.txt` Find the string "stuff" in all the .txt files
-print`
```

```
[vagrant@rhel-cdk ~]$ cd ~
[vagrant@rhel-cdk ~]$ echo "Stuff" > target_file.txt
[vagrant@rhel-cdk ~]$ echo "more" >> target_file.txt
[vagrant@rhel-cdk ~]$ ls
my_directory target_file.txt
[vagrant@rhel-cdk ~]$ grep -i stuff `find . -name *.txt -print`
"Stuff"
[vagrant@rhel-cdk ~]$ []
```

---

<b>head [file]</b>	Output the first part of file (first 10 lines)
--------------------	------------------------------------------------

<b>curl developers.redhat.com</b>	Retrieve the content from <a href="http://developers.redhat.com">developers.redhat.com</a>
-----------------------------------	--------------------------------------------------------------------------------------------

<b>source myenvsetting_script.sh</b>	How to add something to the PATH and make it stick By default a new shell is launched to run a script, therefore env changes are not visible to your current shell.
--------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

```
[vagrant@rhel-cdk ~]$ ls
myenvsetting_script.sh mystuff
[vagrant@rhel-cdk ~]$ cat myenvsetting_script.sh
export MY_STUFF=/home/vagrant/mystuff
export PATH=$PATH:$MY_STUFF/bin
[vagrant@rhel-cdk ~]$ source myenvsetting_script.sh
[vagrant@rhel-cdk ~]$ echo $PATH
/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/vagrant/.local/bin
:/home/vagrant/bin:/home/vagrant/mystuff/bin
[vagrant@rhel-cdk ~]$ []
```

Note: the path uses ":" as a separator vs ";" in the Windows world

---

<b>sudo yum -y install net-tools</b>	"yum" is the installation tool for Fedora, Centos and RHEL. This command installs "net-tools" which has many handy utilities like netstat
--------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------

```
[vagrant@rhel-cdk ~]$ sudo yum -y install net-tools
Loaded plugins: product-id, search-disabled-repos, subscription-manager
Resolving Dependencies
--> Running transaction check
--> Package net-tools.x86_64 0:2.0-0.17.20131004git.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
 Package Arch Version Repository Size
=====
 Installing:
 net-tools x86_64 2.0-0.17.20131004git.el7 rhel-7-server-eus-rpms 304 k

Transaction Summary
=====
 Install 1 Package

Total download size: 304 k
Installed size: 917 k
Downloading packages:
net-tools-2.0-0.17.20131004git.el7.x86_64.rpm | 304 kB 00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
 Installing : net-tools-2.0-0.17.20131004git.el7.x86_64
 Verifying : net-tools-2.0-0.17.20131004git.el7.x86_64
 1/1

Installed:
 net-tools.x86_64 0:2.0-0.17.20131004git.el7

Complete!
[vagrant@rhel-cdk ~]$ []
```

---

**sudo netstat -anp | grep tcp | grep LISTEN** Lists the various in-use ports and the process using it

```
[vagrant@rhel-cdk ~]$ sudo netstat -anp | grep tcp | grep LISTEN
tcp 0 0 127.0.0.1:10443 0.0.0.0:*
tcp 0 0 127.0.0.1:10444 0.0.0.0:*
tcp 0 0 0.0.0.0:80 0.0.0.0:*
tcp 0 0 0.0.0.0:1936 0.0.0.0:*
tcp 0 0 0.0.0.0:53 0.0.0.0:*
tcp 0 0 0.0.0.0:22 0.0.0.0:*
tcp 0 0 127.0.0.1:25 0.0.0.0:*
tcp 0 0 0.0.0.0:443 0.0.0.0:*
tcp 0 0 0.0.0.0:8443 0.0.0.0:*
tcp6 0 0 :::2376 :::*
tcp6 0 0 ::::53866 :::*
tcp6 0 0 ::::10250 :::*
tcp6 0 0 ::::33900 :::*
tcp6 0 0 ::::9101 :::*
tcp6 0 0 ::::52944 :::*
tcp6 0 0 ::::38549 :::*
tcp6 0 0 ::::22 :::*
tcp6 0 0 ::::7001 :::*
tcp6 0 0 ::::1:25 :::*
tcp6 0 0 ::::4001 :::*
tcp6 0 0 ::::32869 :::*
LISTEN 27967/haproxy
LISTEN 27967/haproxy
LISTEN 27967/haproxy
LISTEN 27967/haproxy
LISTEN 13368/openshift
LISTEN 907/sshd
LISTEN 1651/master
LISTEN 27967/haproxy
LISTEN 13368/openshift
LISTEN 13168/docker
LISTEN 13368/openshift
LISTEN 13368/openshift
LISTEN 13368/openshift
LISTEN 14636/haproxy_expor
LISTEN 13368/openshift
LISTEN 13368/openshift
LISTEN 13368/openshift
LISTEN 907/sshd
LISTEN 13368/openshift
LISTEN 1651/master
LISTEN 13368/openshift
LISTEN 13368/openshift
[vagrant@rhel-cdk ~]$
```

---

**sudo netstat -anp | grep 2376** Lists the process listening on port 2376

```
[vagrant@rhel-cdk ~]$ sudo netstat -anp | grep 2376
tcp6 0 0 ::::2376 :::*
LISTEN 13168/docker
[vagrant@rhel-cdk ~]$
```

---

This is particularly useful when another process is hanging out on a port you need, like if you started Apache on 80 or Tomcat on 8080.

**wget https://someurl.com/somefile.tar.gz**

wget is a useful utility for downloading files from any website. If installation is required, simply sudo yum -y install wget

```
[vagrant@rhel-cdk ~]$ sudo yum -y install wget
Loaded plugins: product-id, search-disabled-repos, subscription-manager
Resolving Dependencies
--> Running transaction check
--> Package Wget.x86_64 0:1.14-10.el7_0.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
wget x86_64 1.14-10.el7_0.1 rhel-7-server-eus-rpms 546 k

Transaction Summary
=====
Install 1 Package

Total download size: 546 k
Installed size: 2.0 M
Downloading packages:
wget-1.14-10.el7_0.1.x86_64.rpm | 546 kB 00:00:00

Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
 Installing : wget-1.14-10.el7_0.1.x86_64 1/1
 Verifying : wget-1.14-10.el7_0.1.x86_64 1/1

Installed:
 wget.x86_64 0:1.14-10.el7_0.1

Complete!
[vagrant@rhel-cdk ~]$
```

---

**tar -xf somefile.tar.gz  
tar -xf somefile.tar.gz -C ~/somedir**

Extracts/expands (think unzip) into current directory  
Expands into the "somedir" directory

## RHEL

# Red Hat Enterprise Linux 8

We are delighted to introduce you to Red Hat Enterprise Linux 8. If you're familiar with previous versions of Red Hat Enterprise Linux, you'll find RHEL 8 more intuitive to pick up and use. However, there are a few new features and changes that you'll want to be aware of, so we hope this cheat sheet will help you quickly explore and begin your RHEL 8 application development.

## SIMPLIFIED SOFTWARE PACKAGING AND INSTALLATION

Installing and using RHEL 8 is much easier than previous releases. Previously, there were server, workstation, and desktop variants, but RHEL 8 uses one installation medium for all variants. The RHEL 8 has also been simplified with fewer repos - they are:

**BaseOS** - primarily core operating system packages with support for the lifetime of the OS

**Appstream** - user-space applications and components, including numerous Application Streams (see below)

**CodeReady Builder** - additional libraries and tools for developers

**Supplementary** - 3rd party support only

Compilers, runtimes, web/database servers, and development tools will generally be delivered as Application Streams from the AppStream repo. See below for more info.

## WORKNG WITH CONTAINERS

To enable container management without the need for daemons, Red Hat has [introduced](#) a set of tools for your Linux container application development:

**Buildah** allows you to build a container without any daemon or docker.

**Podman** allows you to manage containers without the daemon dependency it's also docker cli compatible.

### # podman pull

RHEL 8 compatible images can be found [here](#)

### # yum install -y podman

### # alias docker=podman

type to use podman in place of docker

## RED HAT UNIVERSAL BASE IMAGE (UBI)

Derived from Red Hat Enterprise Linux, the Red Hat Universal Base Image (UBI) provides a freely redistributable, enterprise-grade base container image on which developers can build and deliver their applications. This means you can containerize your app in UBI and deploy it anywhere. Of course, it will be more secure and Red Hat supported when deployed on RHEL or Red Hat OpenShift, but now you have more options. There are separate UBI 7 and UBI 8 versions for RHEL 7 and 8, respectively. You can obtain a number of RHEL container images from the Red Hat container [catalog](#).

## BASIC RED HAT ENTERPRISE LINUX COMMANDS

The most basic tasks that you might need after the operating system has been [installed](#) include:

### # yum search string

search for packages matching a specific string

### # yum install package\_name

install a package

### # yum update package\_name

update a package

### # yum remove package\_name

### # yum history undo last

uninstall a package and any packages that depend on it

### \$ yum list all

list information on all installed and available packages

### \$ yum list installed

list all installed packages

### # subscription-manager repos --list

list all available repositories

### \$ yum repolist

list all currently enabled repositories

### # subscription-manager repos --enable repository

enable a repository

### # subscription-manager repos --disable repository

disable a repository

## INTRODUCING APPLICATION STREAMS

RHEL 8 Beta introduces *Application Streams* where we deliver user space packages (e.g. compilers, scripting languages, databases, etc.) on a cadence that makes sense for each package.

In RHEL 8, Applications Streams are mostly packaged as Modules, but a few are non-module RPMs. A module is a set of RPM packages that can or must be installed together. A typical module can contain packages with an application, packages with the application's specific dependency libraries, packages with documentation for the application, and packages with helper utilities. Modules can have one or more streams - different versions of the module.

### Terms and terminology:

**Application Stream (or simply stream)** - refers to content. PHP 7.2 is an application stream. PHP 7.3 is an application stream.

**Module** - is the packaging format. PHP is packaged as a module.

**Module Stream** - different versions of a component packaged as a module. PHP 7.2 is an application stream packaged in a module stream.

**appstream** - is the name of the RHEL 8 repo where you can find Application Streams.

For even more information about Application Streams and modules, see [Introducing Application Streams in RHEL 8](#).

## FINDING AND EXPLORING MODULES

The following are common module commands.

**\$ yum module list**

list all modules

**\$ yum module list installed**

list installed modules

**\$ yum module provides package**

find which module provides a package

**\$ yum module info module**

examine details of a module

**\$ yum module info --profile module:stream**

list packages installed by profiles of a module

**\$ yum module list module**

display the current status of a module

## WORKING WITH MODULES

The following commands must run with administrator privileges. Note also that some operations with modules require changes to many packages.

**# yum module enable module:stream**

enable a specific stream without installing packages

**# yum module install module:stream/profile**

install a specific stream

**# yum module remove module && yum module disable module**

disable a module stream and remove all packages provided by it

## INSTALLING SPECIFIC APPLICATION STREAMS

The following table lists the most interesting Application Streams available in RHEL 8.

.NET Core 2.1	<b>\$ sudo yum install dotnet</b>
Ant 1.1	<b>\$ sudo yum install ant</b>
Buildah 1.5 & Podman 1.0	<b>\$ sudo yum install buildah podman</b>
Clang/LLVM 7.0	<b>\$ sudo yum install llvm-toolset</b>
GCC 8.2 plus complementary tools	<b>\$ sudo yum group install "Development Tools"</b>
GO 1.11	<b>\$ sudo yum install go-toolset</b>
HTTPD 2.4	<b>\$ sudo yum install httpd</b>
MariaDB 10.3	<b>\$ sudo yum install mariadb</b>
Maven 3.5	<b>\$ sudo yum install maven</b>
MySQL 8	<b>\$ sudo yum install mysql</b>
Nginx 1.14	<b>\$ sudo yum install nginx</b>
Node.js 10	<b>\$ sudo yum install nodejs</b>
OpenJDK 11	<b>\$ sudo yum install java-11-openjdk-devel</b>
OpenJDK 8	<b>\$ sudo yum install java-1.8.0-openjdk-devel</b>
PCP 4.3	<b>\$ sudo yum install pcp-zeroconf</b>
Perl 5.26 & 5.24	<b>\$ sudo yum install perl</b>
PHP 7.2	<b>\$ sudo yum install php</b>
PostgreSQL 10.5	<b>\$ sudo yum install postgresql</b>
PostgreSQL 9.6	<b>\$ sudo yum module install postgresql:9.6</b>
Python 2.7	<b>\$ sudo yum install python2 or yum module install python27</b>
Python 3.6	<b>\$ sudo yum install python3 or yum module install python36</b>

## INSTALLING SPECIFIC APPLICATION STREAMS (cont)

Redis 5	\$ sudo yum install redis
Ruby 2.5	\$ sudo yum install ruby
Rust 1.31	\$ sudo yum install rust-toolset
Scala 2.10	\$ sudo yum install scala
Subversion 1.1	\$ sudo yum install subversion
Swig 3	\$ sudo yum install swig
Systemtap 4.0	\$ sudo yum install systemtap
Valgrind 3.14	\$ sudo yum install valgrind
Varnish 6	\$ sudo yum install varnish

## MORE INFORMATION

For more information about RHEL 8, visit the [Red Hat Developer website](#).

Note: if `sudo` isn't enabled for your user ID, see [How to enable sudo on Red Hat Enterprise Linux](#). During system installation, checking the box Make this user administrator enables `sudo` for your user ID.

---

**Authors:** [Mike Guerette](#), Vladimir Slavik

# **WELCOME To: MODULE 9**

## **ALL ABOUT RESUME**

# Resume Workshop

- How to build your impressive resume
- Do's and Don'ts
- Adding keywords
- Structure.

# **Job Titles and Duties**

- IT technician
- Help desk technician
- Systems Administrator
- Systems Engineer

# COVER LETTER

- Cover letter – Entry Level
- Cover letter – Experienced Level

# EXPOSURE TO OTHER TECHNOLOGIES

- Operating System
  - Windows, Virtualization technology
- Hardware
  - Dell and HP
- Monitoring Tools
  - Nagios, Splunk, Zenoss
- Cloud
  - Amazon Cloud
- OS Tools
  - Redhat Satellite, Kickstart, Active Directory, DNS, Puppet etc
- Database
  - SQL, Informix etc.

## **Job Related Tasks as System Administrator**

### **Administration**

- Install Operating system
- Manage hardware, firmware and operating system
- Services installation and configuration
- Upgrade Linux servers
- Migrate OS from older hardware to newer hardware
- Patch management
- 3<sup>rd</sup> party software installation and configuration

### **Different Type of Issues or Troubleshooting**

- Server is running slow, it take a long time to login
- Disk space is full
- Need more disk space
- NTP is not working
- Server cannot send email Sendmail is not working
- Cant mount NFS file system or CIFS share
- Cannot restart a service
- Cannot reach a server (network related)
- Unable to resolve a hostname from DNS
- Can't get to the DNS server
- Cant download a new rpm package from redhat.com using yum command
- Log file is too big
- Server loses IP every time it reboots
- Filesystem is corrupted
- Rsyslog is not sending logs to the central logger
- Cant mount samba share
- Samba share is very slow
- Cannot SSH to a server
- Cannot copy files to a directory, permission denied
- Cannot login as root using SSH protocol
- User cannot write to his/her home directory. User home directory is full
- User cant login through username and password
- Incorrect hostname
- Memory is high or full
- Login issue with username and password
- Writing scripts to automate tasks
- Backup is not working
- Setup cronjobs to schedule tasks

**Documentation**

- Create and update all documentation

**Meetings:**

- Meet with the vendors or management and keep them updated on the progress of the job/tasks

**Additional Tools:**

- Monitoring tools
- Documentation tools
- Ticketing system
- Timesheet system
- Configuration tools (Redhat Satellite, Puppet, kickstart etc.)

**Vendor Relation:**

- Redhat
- HP
- Dell
- Cisco
- Sun/Oracle
- VMWare
- 3<sup>rd</sup> Party vendors (Application related)
- Microsoft

**3<sup>rd</sup> Party Applications Support:**

- Any application that will be running on Linux servers

**Licensing:**

- Keeping track of all licensing

**Other Job Related Tasks:**

- Being on-call
- Recommendation or introduce new technology
- Attend training or job related seminars
- Project management
- Evaluate 3<sup>rd</sup> party tools

## **BASIC RESUME TIPS:**

- **Include All Your Contact Information**

**Jane Doe**  
82 Mill Lane  
Stilltown, NY, 12170  
Cell Phone: 555-555-5555  
Home Phone: 555-222-2222  
Email: name@ycaprr.com

It's important to include all your contact information on your resume so employers can easily get in touch with you. Include your full name, street address, city, state, and zip, home phone number, cell phone number, and email address.

- What to Include in a Resume Contact Section
- Resume Contact Section Example

- **Add a Profile or Objective**

[REDACTED]  
in a position at AB company where I can maximize my management skills, and my quality assurance, program development, and training experience.

If you include an objective on your resume, it's important to tailor it to match the job you are applying for. The more specific you are, the better chance you have of being considered for the job you are interested in, or consider using a resume profile, with or without a headline, instead.

- **Include Resume Keywords**



Your resume should include the same keywords that appear in job descriptions. That way, you will increase your chances of your resume matching available positions - and of you being selected for an interview. Also include keywords in your cover letter.

- Resume Keywords
- How to Include Keywords in Your Resume
- List of Resume and Cover Letter Keywords

- **Prioritize Your Resume Content**



It's important to prioritize the content of your resume so that your most important and relevant experience is listed first, with key accomplishments listed at the top of each position.

- How to Prioritize Your Resume Content

- **Write a Custom Resume**



It definitely takes more time to write a custom resume, but, it's worth the effort, especially when applying for jobs that are a perfect match for your qualifications and experience.

- How to Write a One Page Resume
- Targeted Resume Writing and Samples
- Lists of Skills for Resumes

- **Tweak for Technology**



In this competitive job-seeking environment, job seekers need to make sure that their resume stands out from the pack, is selected by talent management systems, and shows, in a professional, no-nonsense way, that the applicant has taken the time and interest to pursue a specific job opening.

- Resume Writing Tips for a Technology-Savvy World

- **Choose the Right Resume Format**



There are several basic types of resumes used to apply for job openings. Depending on your personal circumstances, choose a chronological, a functional, combination, or a targeted resume. Take the time to customize your resume - it's well worth the effort.

- Types of Resumes
- Types of Resume Examples

- **Use a Resume Template**



Use a resume template as a starting point for creating your own resume. Add your information to the resume template, then tweak and edit it to personalize your resume, so it highlights your skills and abilities.

- How to Use a Resume Template
- Resume Templates
- Microsoft Word Templates for Resumes and Letters

- **Email Your Resume**



When you're sending an email resume, it's important to follow the employer's instructions on how to submit your cover letter and resume. The employer may want your resume attached to the email message and sent in specific format, typically as a Word document or a PDF.

- How to Email a Resume

## Sample Cover Letter – Entry Level

Imran Afzal

100 No Such Place Ave, New York, NY 10000

[Imran\\_ny@hotmail.com](mailto:Imran_ny@hotmail.com)

917-000-0005

Vandelay Industries

25 Some Wannabe Ave, New York, NY 00001

Dear Mr. Russell,

Please accept the attached resume for the position of System Administrator as recently advertised on indeed.com. I've had the pleasure to review the company's website and went over the job description, and am very enthusiastic about the position. I believe that my combination of technical skills, and business sensibilities would serve Vandelay industries well in this position.

I know I don't have the exact number of experience that you are looking for but with the time spent on my lab and helping out others with technical issues can be quantified for up to 2 years of actual corporate level experience. I am the "go to guy" for everyone at my current job, and among my family and friends.

[HERE] Write about your biggest achievement in your career

My resume will speak about my technical skills but here is a quick summary of additional skills that I am really good at.

- Strong interpersonal and communication skills
- Keen attention to detail
- Analytical skills
- Time management skills
- Excellent documentation skills

I am very much interested in speaking with you to discuss the value I can bring to Vandelay industries. I can be reached at the above telephone number or email address. I look forward to hearing from you.

With best regards,

Imran Afzal

# Sample Cover Letter – Experienced

Imran Afzal  
100 No Such Place Ave.  
New York, NY 10000  
[Imran\\_ny@hotmail.com](mailto:Imran_ny@hotmail.com)  
917-000-0005

Vandelay Industries  
25 Some Wannabe Ave  
New York, NY 00001

Dear Mr. Russell,

Please accept the attached resume for the position of System Administrator as recently advertised on indeed.com. I've had the pleasure to review the company's website and went over the job description, and am very enthusiastic about the position. I believe that my combination of technical skills, experience and business sensibilities would serve Vandelay industries well in this position.

Attached resume includes all the details about my career and education background but here is a quick summary about myself that will highlight how I can be the perfect match.

At my current position as a Help desk technician I build, manage and troubleshoot hundreds of servers. I provide first/second level support to thousands of internal or external customers. My professional experience includes diagnosing and addressing complex hardware and software issues. Also I maintain the highest level of system security and educate users on the latest system updates and technologies. I have saved millions to my company by introducing XYZ.

I am very much interested in speaking with you to discuss the value I can bring to Vandelay industries. I can be reached at the above telephone number or email address. I look forward to hearing from you.

With best regards,  
Imran Afzal

## **VMWare**

VMware, Inc. is a U.S. software company that provides cloud and virtualization software and services, and was the first to successfully virtualize the x86 architecture. Founded in 1998, VMware is based in Palo Alto, California. In 2004 it was acquired by and became a subsidiary of EMC Corporation, then on August 14, 2007, EMC sold 15% of the company in a New York Stock Exchange IPO. The company trades under the symbol VMW

### **What is VMWare?**

The hardware of today, regardless of processor count or core count, was designed to run a single operating system. This leaves most machines vastly underutilized. VMware virtualization lets you run multiple virtual machines on a single physical machine, with each virtual machine sharing the resources of that one physical computer across multiple environments. Different virtual machines can run different operating systems and multiple applications on the same physical computer. VMware is an operating system that sits directly on the hardware and is the interface between the hardware and the various operating systems. It expands the hardware, from the user's point of view, to many different independent servers all with their own processors and memory. These virtual servers cannot be distinguished from physical servers by the end users.

### **How Does VMware work?**

VMware works by loading a small, efficient operating system, or **hypervisor** directly on the host hardware. The VMware hypervisor has a small footprint and is extremely efficient, with a very small (1%) overhead. Device drivers for nearly all major brand devices are available from VMware. These are loaded during the configuration process.

The client operating systems, such as Microsoft® Server 2008, Linux varieties, etc. are then set up as virtual machines, working directly with the VMware layer rather than with the actual hardware. This allows replacement of hardware to be very simple. If the hardware is replaced, VMware is reconfigured for the new hardware, and the virtual guest operating systems see no change whatsoever and are immediately able to boot and operate.

### **How Resources Allocated? Are Separate Cores Needed for Each O/S?**

#### ***Processors***

To explain how processor resources are allocated, first of all one need to understand how VMware treats processors. As an example, consider a server, with 128GB of memory and Dual 8-Core 2.7Ghz Intel® Xeon™ processors. This would be presented to VMware as a processor pool at  $(16 * 2.7) = 43\text{Ghz}$ . During the configuration process for virtual operating systems guests, each VM can be assigned a portion of that available processor pool. This assignment is most often left to VMware to perform dynamically during operation.

#### ***Memory***

Memory usage can also either be set during configuration or left to VMware to configure dynamically. You may wish to limit memory usage on some servers, such as exchange, that will attempt to consume as much memory as is available.

#### **Data Storage and Drives**

VMware is capable of using a drive pool located physically within a server. Regardless of how the storage

is physically accomplished, client operating systems are set at configuration to use a portion of the drive pool for their 'boot', or 'C' drives. Other storage can be set up logically so as to available for access by any of the virtual machines. Again, to users of those machines, the drives may set to appear as physical drives attached to the virtual machine.

### **What is required for VMware?**

Any server is capable of acting as a physical VMware host. The speed and core count of the processors, or, as defined above, the processor pool, should be matched to the sum of requirements of the virtual operating systems that will be installed. The required memory capacity is also a function of the requirements of the virtual clients.

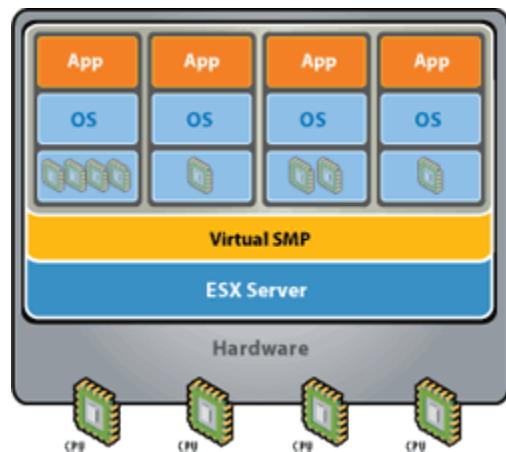
### **Benefits**

VMware allows the enterprise to replace many disparate, underused devices with a few virtual hosts. This greatly reduces system downtime, allows for simple movement of virtual clients from one hardware host to another and allows for scheduled hardware repair or replacement with downtime by moving those clients to another hardware host on the cluster. It also allows the IT administrator to very quickly add virtual servers as required without the need to purchase additional hardware. Upgrading hardware becomes a simple process. Removing the requirement of the Operating System needing to work directly with the hardware makes disaster recovery or replacement of failed servers simple

### **VMWare Major Components:**

#### ***Virtual Machine***

A VM is a software computer that like a physical computer runs an operating system and applications. An operating system installed on a virtual machine is called guest operating system. Because every VM is an isolated computing environment, you can use virtual machines as desktop or workstation environments as testing environments or to consolidate server applications. In vCenter server, VM run on hosts or clusters. The same host can run many VMs



## *ESXi*

It is the name of Operating System utilized and built by VMWare. It is also referred as a host sever. A host is a computer that uses virtualization software, such as ESX or ESXi to run virtual machines. Hosts provide the CPU and memory resources that virtual machines use and give virtual machines access to storage and network connectivity. Once installed and configured then you would not be required to have configuration accessed. It will be accessed through vSphere client

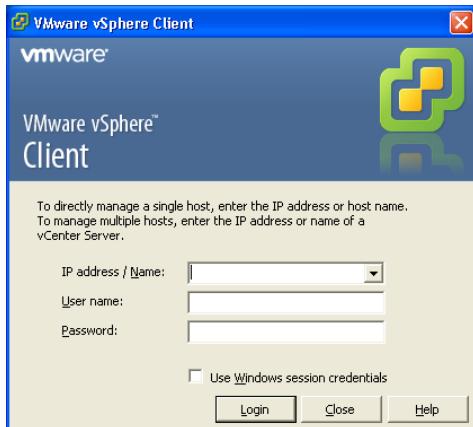
ESXi Configuration Screenshot:

System Customization	Configure Management Network
<p><b>Configure Password</b> <b>Configure Lockdown Mode</b></p> <p><b>Configure Management Network</b> <span style="background-color: black; color: black;">Restart Management Network Test Management Network Network Restore Options</span></p> <p><b>Configure Keyboard</b> <b>Troubleshooting Options</b></p> <p><b>View System Logs</b></p> <p><b>View Support Information</b></p> <p><b>Reset System Configuration</b></p>	<p><b>Configure Management Network</b></p> <p>Hostname: localhost</p> <p>IP Address: 192.7.100.129</p> <p>Network identity acquired from DHCP server 192.7.100.254</p> <p>IPv6 Addresses: fe80::20c:29ff:fea:87d4/64</p> <p>To view or modify this host's management network settings in detail, press &lt;Enter&gt;.</p>

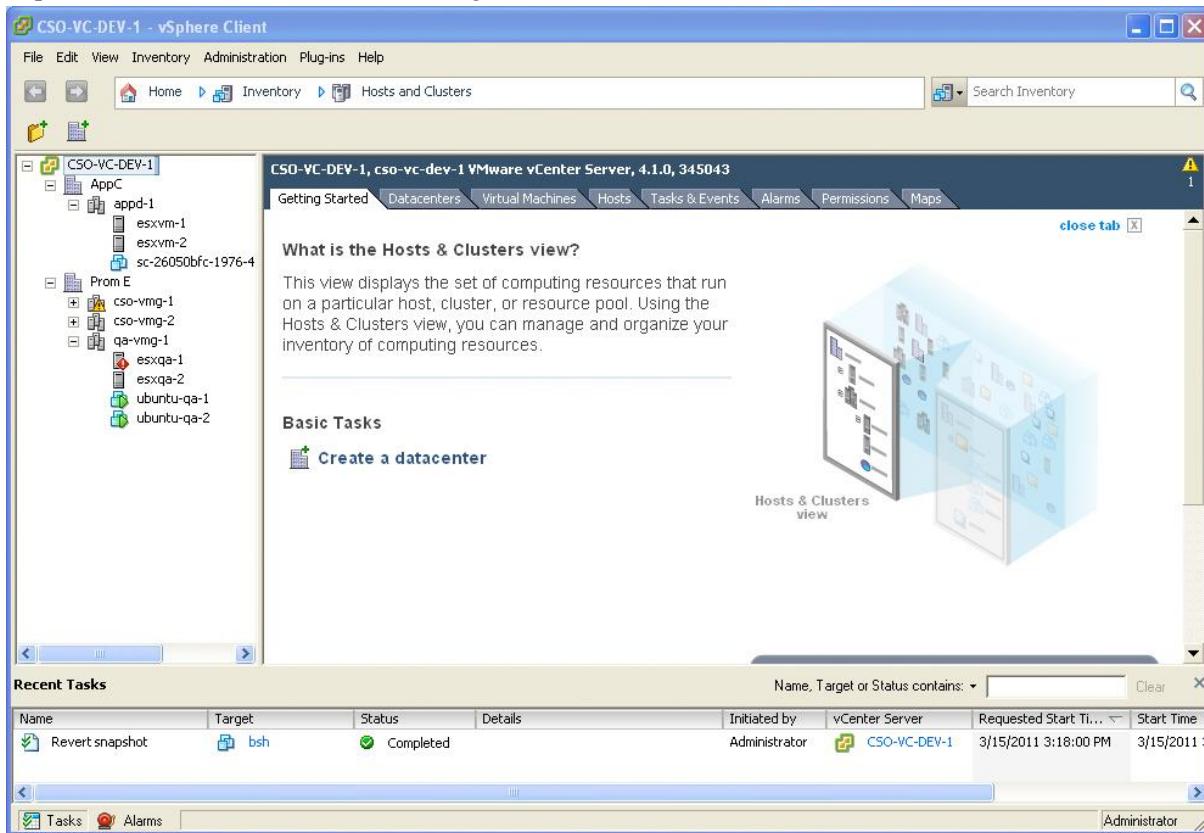
## vSphere

It is a client which allows you to connect to hosts or vCenter server. It works just like putty or RDP client where you would specify the IP, username and password to login to the host or vCenter server  
There are 2 types of vSphere clients, thick and thin client. The thick client has to be installed on the Windows machine where as thin client is a web-based client and can be accessed from any machine running a web browser

vSphere Thick Client Login Screenshot:



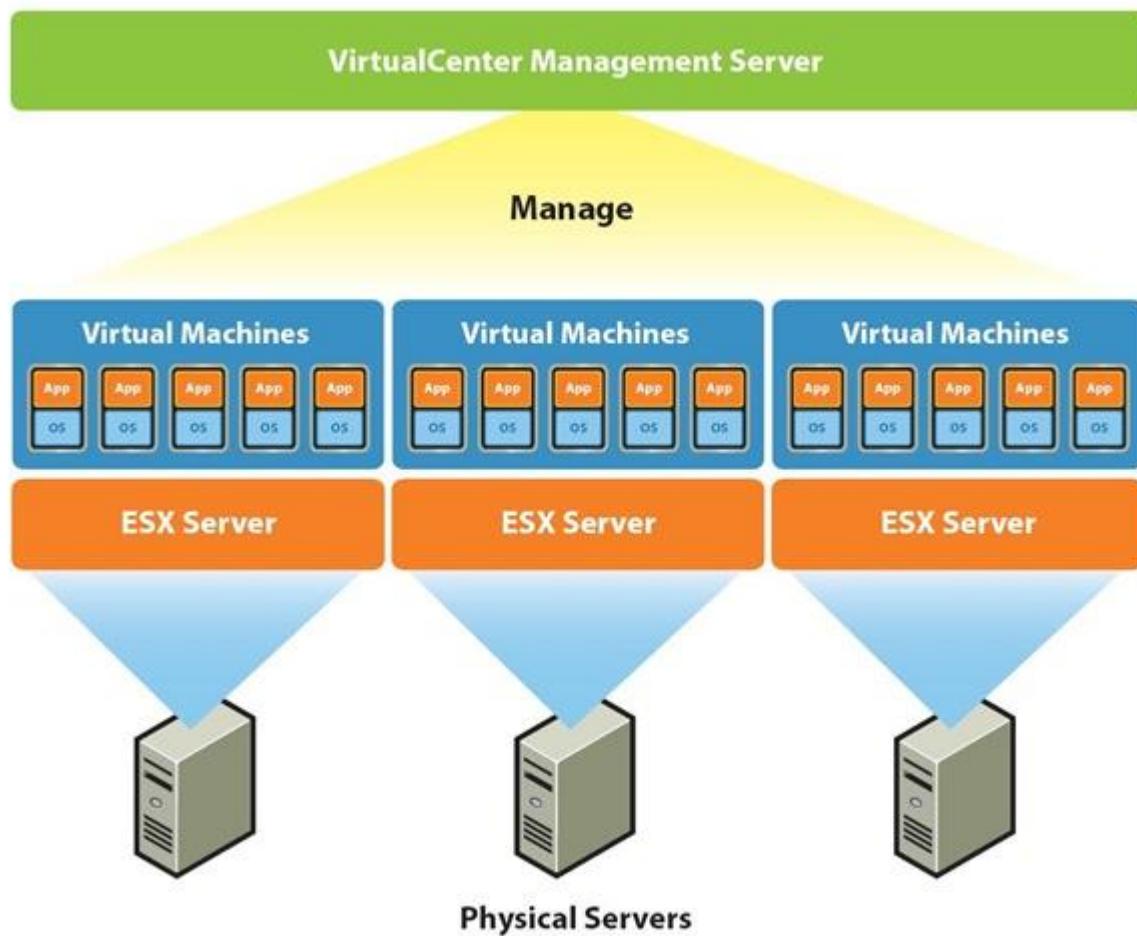
vSphere screenshot after a successful login:



## *vCenter*

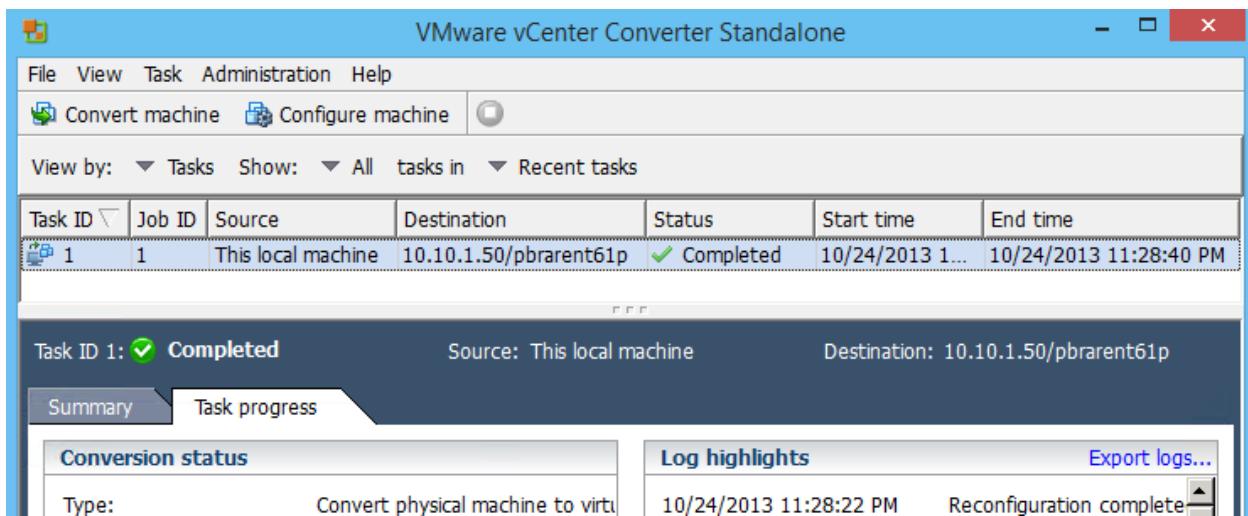
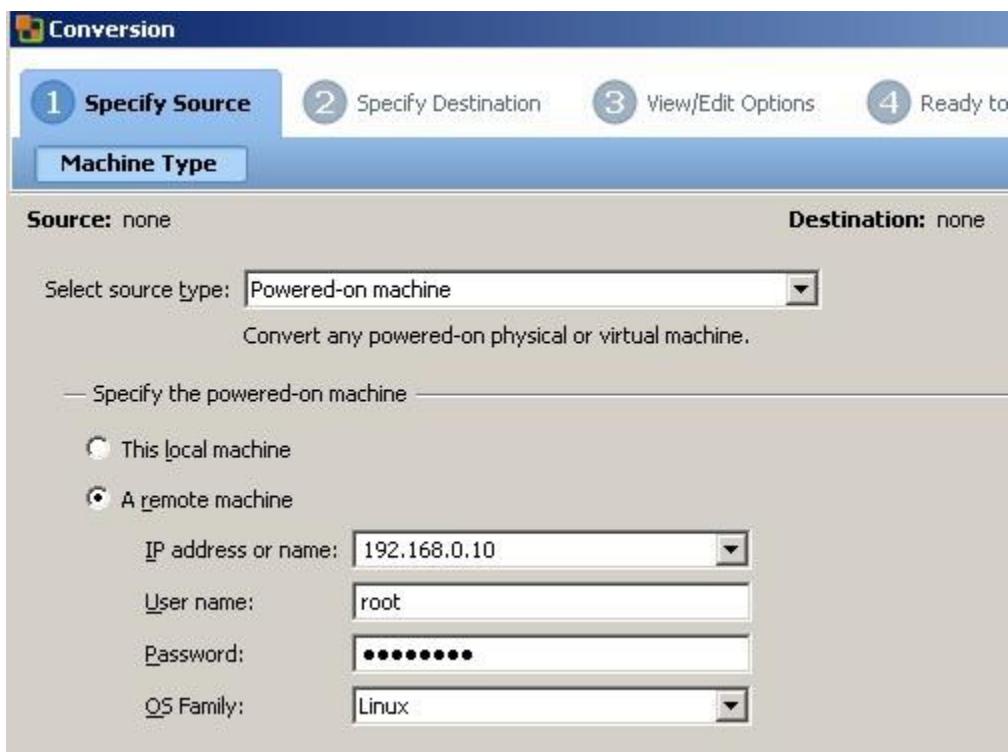
This VMWare component is used when setting up ESXi hosts in a cluster environment. The vCenter can be accessed the same way as ESXi server through vSphere client. Once the vCenter server is installed and configured then all ESXi servers can be added and a cluster can be configured. There are 2 types of vCenter servers, Windows based and Linux based. For Windows, you would need to install the executable on Windows server. For Linux, you would download the Linux vCenter appliance which will be a Virtual machine itself

vCenter management layout:



### *vCenter Converter*

It allows you to convert physical server to virtual, virtual to virtual and virtual to physical server. This components runs on window environment



### **VMWare Initial Configuration:**

- Install ESXi software on bare-metal hardware.
  - Insert the CD/DVD into the hardware
  - Go to BIOS or hit F8 on your server to boot from the ESXi CD
  - Once the ESXi installation screen appears then simply follow the wizard which is self-explanatory or watch the video on youtube on “ESXi 5.5 installation”  
<https://www.youtube.com/watch?v=UVv5K8RItYc>
  - During the installation process, make sure to configure the network, username and password
- Install vSphere client on your Windows laptop or use vSphere web client (*for vCenter*)
  - On your Windows laptop/workstation, install the vSphere thick client and just simply follow the wizard prompt. Make sure the vSphere client has to be the same version as ESXi
  - You can also open up a web-browser and type in the IP address of ESXi server which was configured during the ESXi server installation
  - It will prompt you to HTTPS secure session, click confirm certificate and continue
  - On the “VMware ESXi 5 Welcome” screen you will see one of the following options
    - Download vSphere Client (Follow this link and download the vSphere client)
  - Please note that the web client of vSphere only runs on vCenter server
- Once the vSphere client is installed then double click on the vSphere icon and input IP, username and password of ESXi server which will then bring up the GUI of ESXi/vSphere page

Following the links below to view the installation process:

- VMWare ESXi Installation:  
<https://www.youtube.com/watch?v=UVv5K8RItYc>
- VMWare vSphere Client Installation:  
<https://www.youtube.com/watch?v=IzYUXwGXk14>
- Setting up new VM on ESXi server  
<https://www.youtube.com/watch?v=7ZZAS-7bUs8>

# SOMEONE GREAT

Somewhere Hills, NY 10000  
212-000-0000  
[someone@smail.com](mailto:someone@smail.com)

## **OBJECTIVE – Pick Any**

1. Seeking a challenging and rewarding career as a Linux administrator professional where I can contribute my knowledge and skills for growth and development of the organization. I am committed to make a change in my career and I will do anything to achieve my objective. I believe everyone deserves a chance to improve his/her life. I might not have the right number of experience but what I have is sincerity, dedication and passion to work hard. So I will make sure I will leave no stones unturned to satisfy my superiors and get the job done.
2. Hard working, dependable Linux System Administrator seeking position within a growing or strong organization today
3. As a Linux Administrator, my objective is to strengthen the company's applications and system features, configure servers and maintain networks to reinforce the company's technical performance. Develop tools needed for effective database setup, ensure that Linux environment is performing at its optimum level, manage system backups and provide infrastructure support. To be able to fulfill this, I will utilize the knowledge and expertise I have gained in my years of experience as Linux Administrator I am confident that this objective is possible
4. I am a mid-level Linux System Administrator with 4 years of experience in Linux environment and over 5 of experience in windows looking for an opportunity to further enhance my skillset in administering and maintaining systems. I am highly proficient with configuring/managing user workstations and troubleshooting hardware/software/network issues. With my existing knowledge and expertise, I'm looking to obtain a challenging position associated with new technology
- 5.

## **Education**

**Goodwill College                    2002 - 2004**  
Assoc. of Science  
Major: Computer Systems and Technology

## **CLASSES/COURSES**

Professional Linux Training:

- Complete Linux Training Course – [udemy.com](https://www.udemy.com)

## **AWARDS/RECOGNITIONS:**

- Employee of the Month
- “The Extra Mile” award
- Shoutout Employee
- Core value recognition
- Quality Service Recognition (SQR)
- Runner Up award

## **TECHNICAL SKILLS:**

**Operating System:** CentOS/Redhat, Windows, VMWare, Oracle Virtual Box

**Computer Languages:** Linux scripting in bash

**Networking:** Static IP assignment, NIC management, FTP, NFS, NTP/Chronyd, HTTPD, DNS/Bind, rsyslog etc etc.

**Database:** Exposure to Microsoft Access or MySQL etc.

**OS Applications:** Microsoft Word, Excel, PowerPoint, Linux Sendmail, DNS, SSH, SCP, etc

## **HIGHLIGHTS OF QUALIFICATIONS:**

**SOMEONE GREAT**  
Street Address, City, ST ZIP Code  
212-000-0000  
[someone@smail.com](mailto:someone@smail.com)  
[Website](#)

- Remarkable experience in automating rapid provisioning
- Huge knowledge of managing large server environments
- Deep knowledge of Symantec Netbackup and virtualization technology
- Solid understanding of Linux and web server security
- Familiarity with SAN & NAS technology
- Amazing ability to implement projects
- Excellent customer service and interaction skills

#### **Experience**

- Install and configure Redhat/CentOS linux on virtual environment
- Experience in installation, configuration, support of Linux (Redhat) and Windows operating systems
- Provide technical support for both hardware and software issues for end-users
- Create files and directories at user and root level
- Manage multiple operating systems and technologies like Linux, Windows and SQL database
- Alert management from the enterprise monitoring tools, escalations from IT Service Desk
- Extensive experience of addressing Linux systems operations center incidents, reporting, analyzing and resolving production issues
- Perform incident analysis like process management, CPU and memory analysis to quickly recover from service interruptions, and to prevent recurring of issues
- Filesystem management, system upgrade, software patches using yum and rpm tools
- Experience with logical volume management (LVM), analyzing the disk usage and create disk partitions
- Respond to high volume of support tickets reported by service desks
- Working knowledge of virtualization (VMWare)
- Experience in file system maintenance like responding to file system alerts, finding the file system usage and analyzing the highest offender and increasing file systems space
- Managing, installing, and, upgrading software packages using YUM and RPM and creating YUM repository files for offline servers
- Perform and manage system backups and replications running on 3<sup>rd</sup> party backup solutions and through local RHEL tools (e.g. tarball, gzip, dd etc.)
- Work with Service Desk and other Support teams to quickly resolve high impacting incidents
- Monitor system logs, performance monitoring and capacity planning in anticipation of system resource usage and needs
- Monitored systems, networks and infrastructures using tools such as SiteScope, Zabbix or Nagios
- Support other RHEL versions (5, 6 and 7) running in production and development environments
- Setup and administer user and groups accounts along with setting permissions for users
- Limit user account privileges by configuring sudoers
- Performed kernel tuning based on the applications to be installed and the number of users
- Schedule various regular, periodic, future and queue tasks by using crontab
- Enable system logging and network logging of servers for maintenance
- Work on shell scripts for CPU usage, Memory consumption and number of I/O to collect performance stats and troubleshoot issues on RHEL 5/6/7
- Install and configure SSH key-based authentication
- Managed break/fixes through ticketing solutions

**SOMEONE GREAT**

Street Address, City, ST ZIP Code

212-000-0000

someone@smail.com

Website

- Deploy and manage Linux VMs on ESXi server using vSphere client
- Install, administer and troubleshoot NFS, DNS, and FTP servers
- Provide ongoing support on networking services and protocols such as TCP/IP, NIS, DNS, NFS, LDAP, FTP, SSH, SNMP, SSL, and HTTP
- Implementation and Integration of NTP for time synchronization (`ntp` and `chronyd`)
- Perform server refresh and maintain server integrity by configuring firewall rules for network communication
- Configure dynamic and static network settings for IPv4 and filter packets
- Install, configure and manage Apache (HTTPD) on Linux production servers
- Perform error analysis and recovery, documentation and procedures required for reference
- Provide 24x7 on call duties on a rotation basis with the other system engineers as part of incident management
- Participate in Change Control Management and documentation of the procedures
- Experience in decommissioning of servers and monitoring inventory
- Install and configured RAID 0/1/5 at hardware level
- Coordinate with Application support, Server Operations, Network Operations, Middleware and Database Support teams in troubleshooting production issues
- Communicate system status and coordinate maintenance activities
- Experience working with SAN storage setup and configurations (Hitachi)
- Maintain systems by keeping them compliant with company standards for optimum performance and capacity utilization
- Report issues and communicated effectively with the IT management
- Perform data center operations task following user guides, operational processes and SOPs
- User files permission management including assigning, adding and updating permissions at the file and directory level
- Provision UNIX accounts, groups, and assignment of accounts to groups
- Initiate and facilitate technical bridges as per businesses need for command center.
- Use help commands for fast and efficient reference (`man`, `whatis`, `-help`)
- Everyday use of filters and text processing commands (`cut`, `sort`, `grep`, `awk`, `uniq`, `wc`)
- Keep the system secure by monitoring user activity through secure log messages and tools such as `who`, `last`, etc.
- System management using commands (`top`, `df`, `dmesg`, `iostat` 1, `netstat`)
- Setup aliases for ease of management
- Manage OS network (`ping`, `ifconfig`, `netstat`, `tcpdump`, `networking config files`)
- Build and manage development and QA environment on cloud services (AWS and Azure)
- Troubleshoot issues with VMs running on VMware ESXi 5.5 and 6.0.
- Troubleshoot and resolve day to day issues using Linux native tools and analytical skills
- Install, configure and maintain Windows OS versions 2012 R2 (or later), including OS patching, drivers and hardware firmware
- Demonstrate engineering excellence in the design, systems management, automation, patching, deployment, troubleshooting, and performance tuning for Linux based operating systems
- Recommend and implement process improvements and automation to maximize system availability and consistency
- Set strategic direction and manage implementation.
- Provide accurate project estimates, while meeting project deadlines
- Provide timely, creative and alternative solutions to meet business needs
- Provide significant weeknight and weekend support
- Display exceptional people skills and maintained calm demeanor during every interaction

**SOMEONE GREAT**  
Street Address, City, ST ZIP Code  
212-000-0000  
[someone@smail.com](mailto:someone@smail.com)  
Website

- Worked closely with executives/Partners and senior-level management in resolving issues
- Write and maintain documentation such as instruction guides, troubleshooting procedures, incident process, SOW, Root cause analysis etc.
- Setup aliases, short scripts and links for ease of management
- Educated users on business interruption risk and received their buy in for system standardization and update.
- Process management for system updates to minimize risk of application failures
- Created and manage Solaris Jumpstart and Linux Kickstart servers and processes to automate and standardize the installation process, reducing installation time by 35% and post-installation errors by 50%
- Document and implement a disaster recovery plan that included backup schedule policies, regular testing, and off-site recovery plans to ensure data integrity and assurance
- Installed and configured LDAP servers to authenticate between Linux, Apple and Windows environments
- Resolve security vulnerabilities by analyzing and recommending improvements in communications and network security at the component level
- Ensure business continuity by designing, implementing and testing disaster recovery systems
- Manage, monitor and test individual and group user access privileges and security
- Eliminated extended downtime of network and processes by installing and configuring a Nagios server to monitor Windows/Linux servers

**Guest Services Associate**

**ABC - New York, NY**

December 2014 to 2015

**TECHNICAL SUPPORT III**

- Assisted customers having hardware, software and networking issues.
- Assisted business customers with LAN /WAN / Static Ip configuration issues.
- Troubleshoot customers issues with home health check / node health check tools for possible outages.
- Troubleshoot any problems or issues that arise with Windows desktops/laptops from the network drop to the desktop/laptop prioritizing systems infected with viruses
- Maintained updated knowledge of company products and services to better provide customer support and service solutions
- Load and reload, configure, and update anti-virus and virus scan software on desktop/laptop as required
- Gathered customer and technology information to determine technical support level; escalated issues to the appropriate department as necessary.
- Assisted team members and provide support and solutions to customer queries to meet company objectives.
- Worked and Assisted various internal projects for company
- Break-Fix hardware and software related issues on desktops/laptops workstation and printers.
- Configured and deploy desktops/laptops for new hires and IMAC service for office moves.
- Created spreadsheet reports regularly as required by manager for asset management, asset information of new hires' devices, etc.
- Assisted regional IT teams with Smart Hands support in the facilities such as IDF/MDF closets, Data centers and other related server rooms.
- Imaging and re-imaging desktops/laptops using USB/network.
- Provide deskside support in person, on the phone and remote using LogMeIn, Bomgar and Skype.
- Utilized Active Directory to add/remove users, unlock accounts and reset passwords.

**SOMEONE GREAT**

Street Address, City, ST ZIP Code

212-000-0000

someone@smail.com

Website

- Resolve tickets and request in in timely manner
- Set up conference room for presentations, physical Video and Audio Conferencing for end users.
- Ordered and replaced necessary parts for user's machines to be in the correct Operational Unit
- Sustain network integrity, server deployment and security
- Actively participate in process improvement initiatives
- Work cross functionally to develop, implement and support new end user solutions
- Assist in migration existing services from physical data centers to AWS cloud

#### **ADDITIONAL INFORMATION**

##### **PERSONAL SKILLS**

- Problem Solving Skills
- Willingness to Learn
- Effective oral and written communication skills
- Strong analytical problem-solving capabilities
- Project management skills



AQ

# First Name Last Name

E. first.last@gmail.com

C. 347-000-0000

City, NY 10000

## OBJECTIVE

Hard working, dependable Linux System Administrator seeking position within a growing or strong organization today

## Education and Training

B.A

Some College

## TECHNICAL SKILLS

### Operating System:

Redhat/CentOS, Ubuntu and Windows

**Computer Languages:** Linux scripting in bash

**Networking:** NIC Bonding, DNS, FTP, SFTP, SSH, SCP, NTP, NFS, TCP/IP, LAN, WAN, Ethernet config

**Database:** Installation and configuration experience in Access, Oracle, Informix, MySQL

**OS Applications:** SVM, LVM, Veritas, Jumpstart, Kickstart, Active Directory, MS office suite, Sendmail, Apache, rsyslog etc.

**Hardware:** HP DLs, Dell R series, etc.

**Storage:** Sun SAN, HP MSA, Hitachi

## Classes/Courses

Complete Linux Training Course - udemy.com

Linux Troubleshooting Course - udemy.com

## Work Experience

ABC

*Linux System Administrator / New York City, NY 01/2015 to Current*

- Perform installation, configuration and management of Linux servers that runs Braintree mobile and web payment applications for e-commerce payment processing.
- Three main applications are Checkout UI, Payment Method Types and Customer Data.
- Manage 3 infrastructure environment, (production, QA and development) consisting of approximately 7000+ virtual machines and 500+ physical servers.
- Log and troubleshoot issues reported by customers such as Uber, Airbnb, Dropbox, Skyscanner, Pinterest, Nokia and PagerDuty.
- Support a mix of HP and Dell hardware running multiple operating systems (Redhat, CentOS, Ubuntu, Windows and some SUSE).
- Solve infrastructure related issues by collaborating daily with peers and other teams in the Braintree ecosystem to answer merchant questions.
- Collaborate via Slack (Braintree internal chat tool) channel to crowdsource answers to tough questions that come from our merchants and to serve as a technical knowledge hub for other teams.
- Management of monitoring tools such as Nagios and SiteScope which covers about 90-95% of infrastructure devices.
- Manage, monitor and test individual and group user access privileges and security.
- Analyzing and interpreting system and application log files.
- Install, configure and manage of services such as DNS, NTP/Chrony, HTTP, NFS, FTP, Sendmail, OpenLDAP etc.)

**Monitoring Tools:** SiteScope, Spiceworks and Nagios

**Virtualization:** VMWare, Citrix Xen, Oracle virtualization

**Change Management/Ticketing:** ServiceNow and ServiceDesk Plus

- Perform incident analysis like process management, CPU and memory analysis to quickly recover from service interruptions, and to prevent recurring of issues.
- Filesystem management, system upgrade, software patches using yum and rpm tools.
- Experience with logical volume management (LVM), analyzing the disk usage and create disk partitions.
- Respond to high volume of support tickets reported by service desks.
- Working knowledge of virtualization (VMWare).
- Write and maintain documentation such as instruction guides, troubleshooting procedures, incident process, SOW, Root cause analysis etc.
- Create and manage Solaris Jumpstart and Linux Kickstart servers and processes to automate and standardize the installation process, reducing installation time by 35% and post-installation errors by 50%.
- Document and implement a disaster recovery plan that included backup schedule policies, regular testing, and off-site recovery plans to ensure data integrity and assurance.
- Install and configure LDAP servers to authenticate between Linux, Apple and Windows environments.
- Improve overall system's performance and reduce cost by introducing new technology.

## XYZ

*Help Desk Associate / New York, NY 01/2014 to 01/2015*

- Break-Fix hardware and software related issues on desktops/laptops workstation and printers.
- Troubleshoot issues that arise with Windows desktops/laptops from the network drop to the desktop/laptop prioritizing systems infected with viruses.
- Maintained updated knowledge of company products and services to better provide customer support and service solutions.
- Gathered customer and technology information to determine technical support level; escalated issues to the appropriate department as necessary.
- Assisted team members and provide support and solutions to customer queries to meet company objectives.
- Created spreadsheet reports regularly as required by manager for asset management, asset information of new hires' devices, etc.
- Utilized Active Directory to add/remove users, unlock accounts and reset passwords.
- Resolve tickets and request in in timely manner.

# **WELCOME TO: MODULE 10**

**ALL ABOUT INTERVIEW**

# **TRUE MEANING OF IT**

**Transfer of information/data from  
one person to another, from one  
location to another through  
Technology**

# COMMON QUESTIONS

## WHAT



Files, Folders, PDFs



Images



Videos



Music



Instant messages



Emails

## WHERE



Person to Person



Location to Location



Company to Company

## WHEN

24  
HOURS  
A DAY

7  
DAYS  
A WEEK

365  
DAYS  
A YEAR

## WHY



Decision



Problem Solving



Planning



Saves Time



Improvement



Saves Money

## HOW



IT Components

# IT COMPONENTS

- Hardware
- Operating System
- Applications/Software
- Networking
- Security

## Enterprise Level IT Components

- High Level Security
- Storage
- Database

# Post Resume and What Expect

- Recruiters or Companies' Human Resources
- Job description
- Full time or Consultant
- Rate negotiation
- Updated resume with rate confirmation
- Quick interview with the recruiter's manager over the phone
- Meet the recruiter in person
- Resume submission
  
- Interview schedule
- First phone interview
- Second in-person interview
- Offer letter
- Acceptance and start date.

# Interview Workshop

- Employer/Manager (10 minutes)
  - Company overview
  - Job overview
  - Interview process
- Tell me about yourself (5-10 minutes)
- Why do you want to leave or what are you looking in your next job (2 minutes)
- Behavior questions (15 minutes)
  - How do you handle a difficult situation
  - How do you meet your goal if you face an obstacle
  - How do you handle difference of opinion
- Technical questions (25 minutes)
- Any questions for us (5 minutes).

# JOIN THE COMMUNITY

- Join IT Community, Forums and Blogs
  - [www.centos.org](http://www.centos.org)
  - [www.Linuxquestions.org](http://www.Linuxquestions.org)
  - [www.linuxforums.org](http://www.linuxforums.org)
  - [www.linux.org](http://www.linux.org)
  - [www.linux.com](http://www.linux.com)
  - [www.howtogeek.com](http://www.howtogeek.com)
- Other social media e.g. facebook, twitter etc.

# Redhat Certifications

- 3 Redhat Certifications

- ✓ RHCSA
- ✓ RHCE
- ✓ RHCA



- Red Hat Certified System Administrator ([RHCSA – EX200](#))

- Linux administration class (Udemy class suffice)
- Linux system administration experience

- Red Hat Certified Engineer ([RHCE – EX300](#))

- Must be RHCSA

- Red Hat Certified Architect ([RHCA](#))

- Must be RHCE
- To attain and maintain RHCA status, an RHCE must pass at least 5 other Redhat certifications



# Redhat Certifications

- 3 Redhat Certifications
  - Red Hat Certified System Administrator (**RHCSA – EX200**)
    - Linux administration class (Udemy class suffice)
    - Linux system administrator
  - Red Hat Certified Engineer (**RHCE – EX300**)
    - Must be RHCSA
  - Red Hat Certified Architect (**RHCA**)
    - Must be RHCE
    - To attain and maintain RHCA status, an RHCE must pass at least 5 other Redhat certifications



## **Interview Workshop**

Now this is time where you have to shine and prove yourself that you qualify for the job. Think of yourself as a car salesman and you have to sell yourself to the hiring manager. What are the things you are proud of, what do you know, what is your education background, what have you done in the past, how do you match for this position. You need to speak of all these features to sell yourself.

### **ENTER →**

- **Employer/Manager (10 minutes)**
  - Company overview
  - Job overview
  - Interview process

Usually the interview last for an hour. You will get a call from the hiring manager or you will be on-site meeting with him (lets call him Russell). Introduce yourself to everyone. Tell them that “my name is Imran Afzal and thank you so much for your time today for this interview”. Then from here the hiring manager (Russell) will introduce himself or others if there are more people on the interview panel. The first 5-10 minutes will be utilized to give you the company overview e.g. What does the company do, how long they have been in the market, what have they achieved. Make sure to research on the company as well before the interview. This will tell Russell that you are interested and have done your homework.

Next thing Russell will tell you about the job itself. He will go over the quick description of the job, what they are looking for, why they are looking for, the number of people in the team and what they will expect from the new person. Make sure to write this part because this information will help you answer according to their requirement. e.g. if Russell mention that we are looking for someone who is also a team player or someone who can improve the documentation etc. then later you can say that you have worked as team or have done a lot of system documentation. If you are in-person then make a mental note about this.

Before Russell starts asking you questions he will mention about the interview process. He will layout how the interview will be conducted. How much time will be spent on the technical questions, how much time will be spent on the behavior questions and then in the end there will be some time allocated for me to ask any questions. Explaining the interview structure is what great hiring managers do to stay on track.

## **ENTER →**

- Tell me about yourself (5-10 minutes)

Ok so all said and done. Now one of the favorite questions every hiring manager asks. "Tell me about yourself". You should take anywhere from 5-10 minutes to answer this question. Now this is the opportunity to sell yourself. You should break down your answer into 5 categories

### **Category 1.**

Start by introducing yourself by saying, My name is Imran Afzal and I am Systems Engineer by profession. My expertise are Linux, VMWare, Windows, technical training, project management etc. so basically list a quick summary of your skills.

### **Category 2.**

I have been working with Art Vandaley industries for the last so n so years as a system administrator. If you are not working in IT then say "I have been working on my Linux lab for the last 6 months along with my full time job as Sales associates in Bloomingdales."

If you are already in IT then go over a quick summary of everything you do. E.g. build servers, managed them, provide 24x7 support, troubleshoot system related issues and so on.

If you are not in IT then really quickly go over your day to day tasks but don't spend more than 20-30 seconds on it. Focus more on the technical side e.g. work on POS systems or sometimes also troubleshoot technical issues. If you don't do any technical work at then don't lie about it. Move on to mention about your Linux lab. How you started and how many hours you spent to learn, how many hours you spend each week to manage it, break it and troubleshoot it different issues.

Next mention your previous job if you had one. Again go over a quick summary of your daily tasks

### **Category 3.**

After your experience tell Russell about your education background. Which college you went to and what was your major. If you majored in accounting or any other concentration aside from Computers then now is the time to tell why you decided to move to IT. Perhaps better future, better job market, you started likely it more when you worked one day with your friend or whatever your reason is.

### **Category 4.**

Now in category 4 bring up your interest. Like... on your spare time you like to play soccer, watch movies, help around the house in washing dishes or cleaning the house. Tell him what you really enjoy doing in your spare time. Once again don't make up things, trust me you will feel proud once you are done with the interview regardless you get the job or not.

### **Next and last category 5:**

Tell Russell why you want to leave your current job to apply for this. Perhaps your contract with the existing company is coming to an end, or you are looking for career advancement where you can apply your technical skills or you simply want to start your career in IT. Then tell him that you have worked as a team player in the past the outcome was excellent. Give a personal example like your friends don't like to go out without you because they like your company. Mention about your documentation skills because remember he brought this point during the initial conversation. Ask yourself what is that one quality you have that your competitor don't and tell that to Russell.

**ENTER →**

- Why do you want to leave or what are you looking in your next job (2 minutes)

Now you have covered this question in your last category 5. Many time hiring manager will include this question with “Tell me about yourself question”

ENTER →

- Behavior questions (15 minutes)

These type of questions are one of the way for Russell to find out your personality whether you are the right fit for this type of environment. It can take anywhere from 15–20 minutes to answer them. He will ask you questions like:

- How do you handle a difficult situation
- How do you meet your goal if you face an obstacle
- How do you handle difference of opinion

Now remember you don't have to be in a corporate environment or to have an IT job to answer these questions. We face these type of challenges almost everyday, with our family, friends, or during a situation which is un-related to any relationship. E.g. you are at the airport and you are given the 2 options either wait for the 2nd flight which will leave 8 hours from now and get you to point B in 2 hours. Or you can rent a car and drive for 6 hours. Which one option is better for you and why picked one on the other. So think of situation like these and answer these questions accordingly.

ENTER →

- Technical questions (25 minutes)

Now let the fun begin. You will be asked technical questions back and forth for 20–25 minutes. Sometimes technical questions are made up of a situation e.g. What if you have to run a script at night but you are not available at night, how would you manage...of course schedule the job with crontab right.

ENTER →

- Any questions for us (5 minutes)

Now once Russell and the team done asking technical questions now its your turn to return the favor. Ask them questions like:

- what is a typical day looks like for this position
- If it is a contract position ask if there is a possibility to become permanent. This will show that you are committed
- How will you measure the success of the person you will hire

These are a few question example you should ask Russell

Now before you say your good byes and shake hands you should say a closing statement. Anything that will stay with Russell and will keep reminding of you when he interviews other candidates. Now what could that be? Well of course it all depends on you and what you are good at. As an example you can say:

- Before I leave I want to say one thing. If you ask me something and I don't know the answer....then I will tell you that I don't know. But I promise you that I will work on it and get the answer for you.
- Before I leave I want to say one thing. I might not be the smartest or most technical person but I one day I will make you proud of your decision for hiring me
- Before I leave I want to say one thing. I might not meet all the requirement of this job but I definitely being loyal to you and will be there whenever you will need me.

Let me tell you a little secret about my career. Long time ago I went for my first job interview. The hiring manager met me and hired me without asking my any technical questions. Now keep in mind I did not have any experience or Linux skills for that matter. A few weeks later after I started my new job I asked my Manager why didn't you ask me any technical questions when I came for the interview? He said because of your attitude. I knew you were hungry to learn and willing to do whatever it takes.

So what I learned that day is...its not what you say... its how you say it



Interviews are nothing if not opportunities to drive yourself crazy.

Just remind yourself to look good, appear confident, say all the right things and don't say any of the wrong ones.

It shouldn't be so hard to follow these guidelines except you'll be on the receiving end of an endless line of questions. Factor in your nerves and you'll be lucky to remember your own name.

### Don't fret.

If you walk into the interview prepared, you can make sure you know what right things to say, and you can stop yourself from saying the following wrong things.

**1. "I hated my last boss."** Your last boss was a miserable person whose main concern was making your life miserable. Of course you don't have a lot of nice things to say; however, don't mistake honesty, which is admirable, for trash-talking, which is despicable.

"If you truly did hate your last boss, I would be prepared to articulate why your last organization and relationship was not right for you," says Greg Moran, director of industry sales and partnerships for Talent Technology Corp. "Then be prepared to explain what type of organization is right for you and what type of management style you best respond to."

### Don't Miss

- [10 best excuses for coming to work late](#)
- [12 things to consider before taking your first job](#)
- [More CareerBuilder.com stories](#)

**2. "I don't know anything about the company."** Chances are the interviewer will ask what you know about the company. If you say you don't know anything about it, the interviewer will wonder why you're applying for the job and will probably conclude you're after money, not a career.

"With today's technology," Moran says, "there is no excuse for having no knowledge of a company except laziness and/or poor planning -- neither of which are attributes [of potential employees] sought by many organizations."

**3. "No, I don't have any questions for you."** Much like telling the interviewer that you don't know anything about the company, saying you don't have any questions to ask also signals a lack of interest. Perhaps the interviewer answered every question or concern you had about the position, but if you're interested in a future with this employer, you can probably think of a few things to ask.

"Research the company before you show up," Moran advises. "Understand the business strategy, goals and people. Having this type of knowledge will give you some questions to keep in your pocket if the conversation is not flowing naturally."

**4. "I'm going to need to take these days off."** "We all have lives and commitments and any employer that you would even consider working for understands this. If you progress to an offer stage, this is the time for a discussion regarding personal obligations," Moran suggests. "Just don't bring it up prior to the salary negotiation/offer stage."

Why? By mentioning the days you need off too early in the interview, you risk coming off presumptuous as if you know you'll get the job.

**5. "How long until I get a promotion?"** While you want to show that you're goal-oriented, be certain you don't come off as entitled or ready to leave behind a job you don't even have yet.

"There are many tactful ways to ask this question that will show an employer that you are ambitious and looking at the big picture," Moran offers. "For example, asking the interviewer to explain the typical career path for the position is fine."

Another option is to ask the interviewer why the position is open, Moran adds. You might find out it's due to a promotion and can use that information to learn more about career opportunities.

**6. "Are you an active member in your church?"** As you attempt to make small talk with an interviewer, don't cross the line into inappropriate chitchat. Avoid topics that are controversial or that veer too much from work.

"This sounds obvious but many times I have been interviewing candidates and been asked about my personal hobbies, family obligations, et cetera," Moran says. "Attempting to develop a rapport is essential but taking it too far can bring you into some uncomfortable territory."

**7. "As Lady Macbeth so eloquently put it..."** Scripted answers, although accurate, don't impress interviewers. Not only do they make you sound rehearsed and stiff, they also prevent you from engaging in a dialogue.

"This is a conversation between a couple humans that are trying to get a good understanding of one another. Act accordingly," Moran reminds.

**8. "And another thing I hate..."** Save your rants for your blog. When you're angry, you don't sway anybody's opinion about a topic, but you do make them like you less. For one thing, they might disagree with you. They also won't take kindly to your bad attitude.

"If you are bitter, keep it inside and show optimism. Start complaining and you will be rejected immediately," Moran warns. "Do you like working with a complainer? Neither will the interviewer."

## INTERVIEW TIPS

### *OVERVIEW:*

Advance preparation for the interview is KEY to calming down the “jitters” before the actual interview. It is normal to be nervous for the interview but lack of preparation will only enhance that tense, uneasy feeling. Please review the below to insure you perform at your peak for each and every interview. Remember to NEVER take any interview for granted as it may result in a strong, future contact even if the job you are interviewing for is not the ideal fit.

1. Researching a company (and a specific business unit) is a critical first step in preparing for an interview. Review the web-site, perform an internet search using Google and review Bloomberg/Reuters for recent company news. It is vitally important to be able to respond confidently to the question: "Why would you want to work for our company and/or our business unit?"
2. Proper business attire is very important which include a dark grey or blue suit with solid white or light blue shirts/blouses. In a corporate environment, the following is usually frowned upon: long hair (on men), goatees (on men), ear rings (on men), nose rings, etc. Please be sure to use your judgment and always put your best foot forward. Additionally, remember to bring an extra copy of your resume including 2-3 professional references from a direct supervisor/manager and an attaché or briefcase to carry your professional belongings.
3. Arrive to the interview 10-15 minutes early – NEVER be late for an interview. Be sure to have all interview contact information readily accessible in the event there is a mix-up or you’re running late. In circumstances beyond your control, first call the interviewer directly to advise that you are running late; and then, secondly, call your Recruiter to give us an opportunity to call the employer, as well.
4. Upon initial greetings, always give a firm handshake, warm smile and direct eye contact. Please try to avoid excessive hand movements, poor eye contact and slouching in your seat. Integrate your personality and don’t be afraid to smile since you will most likely be working with this person in close proximity. Both your skill sets and personality fit can be equally important to the interviewer.
5. Selling your skills is as important as listening to what the interviewer is looking for in his/her ideal candidate. Be prepared to sell your SRA (skills, responsibilities and 3-5

accomplishments) but it is critical you know which of those SRAs are requirements of the role. It is recommended that you ask questions that serve more as “information gathering” rather than “preferential”: i.e. “How much of the role involves accounting work versus the analytical component” (information gathering) rather than “I’d love to get more involved with the analysis” (preferential question).

Avoid over talking and instead focus on concise, clear, direct answers to the Interviewers’ questions. Also, remember NOT to over-talk the interviewer and wait until the appropriate time to respond to all questions/inquiries. Always make sure to answer questions honestly and never put information on your resume that you can’t back up on the interview. First impressions are critical and you don’t want to create any false perceptions of your capabilities.

#### *INTERVIEW QUESTIONS “You May Be Asked”:*

- Please give us examples of how you handled an adverse situation (i.e. tough client) or a stressful deadline?
- What attributes or skills separates you from other professionals in your field?
- List 3 of your biggest accomplishments and 1-2 weaknesses (or areas requiring improvement). Be careful with the “weakness” question; ask your recruiter how to handle.
- Why are you looking to leave your current position and what are you looking for in your next role?
- What do you know about our company and why are you interested in this position?
- What have you liked and disliked about your current & previous positions? (Remember to stay positive; negative comments about former employers are frowned upon)
- What are your short-term and long-term career goals?
- Do you have any supervisory experience? If so, is that a requirement of your next position?
- Tell me the biggest challenge in your career to date.
- Is there a particular industry or company size preference you have? For example, small versus large firm and tell me the advantages of each.
- Please explain your job moves and if applicable, why you were laid off or stayed less than a year in any position?

- How long have you been looking for a new job? If applicable, why do you feel it is taking so long for you to find the right position?
- How best would your supervisor/manager and staff (if applicable) describe you as a professional?
- What are you looking for in your base salary and/or total compensation? How did you arrive at these numbers? (Best answer: "My main focus is finding the right opportunity and a Fair Market Offer. I'd also prefer to have your recruiter negotiate on my behalf and leave you with the fact that I'm extremely interested & excited about this position and your company.")
- Do you have any questions about the position or the company that I can answer for you? (Note, it is critical that you have relevant questions for the interviewer – see "next section.")

*INTERVIEW QUESTIONS "You Should Ask"*

- I have a solid overview of the position; can you tell me more about the specific skills and responsibilities that are keys to my success in this role?
- How would you describe your ideal candidate?
- How would you describe a typical day for this role? What are some of the daily, weekly & month-end deadlines that I need to be aware of?
- Which groups will I be interfacing with and have exposure to on a regular basis?
- What has been the short and long-term career path for others that have been in this or similar positions within your company?
- Can you tell me more about your background and other members of the group?
- What do you like most about your current role?
- Is there anything else in my background that I can expand upon?
- **IMPORTANT:** Do not ask any questions regarding Compensation, Hours or Benefits.

*INTERVIEW "STUFF TO ALWAYS AVOID":*

- NO backup copies of your resume; remember to always bring multiple copies
- Poor attention detail including an incomplete, inaccurate or sloppy application
- Arriving to an interview late and without a preemptive call to notify interviewer
- Unprofessional attire including jeans or casual clothes
- Poor eye contact & a weak, limp handshake
- Discussing compensation, hours & benefits INSTEAD of focusing on the opportunity
- Negative portrayal of past employers and/or job responsibilities
- Over-confident or lacking confidence – both can "hurt" your chances

- Indirect/vague answers to questions as opposed to direct/concise answers
- Lack of short term and long term career goals that are in sync with the responsibilities of the position
- Lacking any relevant questions for the interviewer; this is often interpreted as a lack of interest or motivation in the role
- Overanxious to bypass current role for future opportunities..."not willing to pay your dues"
- Too lackadaisical; interviewing only to get a better sense of market value OR to draw better compensation from current employer in the form of a "counter-offer"
- Obvious lack of interest or enthusiasm in the company or the role
- Unable to take criticism or responding harshly to aggressive questioning from the interviewer i.e. low GPAs or a lot of job moves should be accompanied by a concise, non-evasive explanations
- Inability to handle pressure; appearing too nervous during the interview. Remember, some nervousness is healthy BUT don't allow it to prevent you from performing at a high level during the meeting

## **BEST QUESTIONS TO ASK IN THE INTERVIEW**

1. What would you consider to be the most important aspects of this job?
2. What are the skills and attributes you value most for someone being hired for this position?
3. Could you describe a typical day or week in this position? The typical client or customer I would be dealing with? The expectations To determine how and when you will evaluated, Payne recommend advises asking:
4. What are the performance expectations of this position over the first 12 months?
5. What types of skills do you NOT already have onboard that you're looking to fill with a new hire? The department asking about your department's workers and role in the company can help you understand more about the company's culture and hierarchy.
6. What is the overall structure of the company and how does your department fit the structure?

## Main – Post Resume and What to Expect

Lets talk about what will happen when you post your resume online e.g. indeed, career builder, zip recruiter, monster or whichever website you prefer.

- **Recruiters or Companies' Human Resources**

**ENTER →** Your resume will be searched by the HR or the recruiters based on the resume title or the keywords inside of your resume.

Now who are recruiters and HR? Good question. Recruiters are also referred as headhunters and they work independently to find the right candidate for the right job. Sometimes they have exclusive relationship and contracts with the hiring companies to find them a person for an open position. Companies prefer to work with the recruiters because they specialize in this field. Recruiters search for all the candidates online, shortlist them based on their skills, academic background or experience. They perform initial interview, set the expectation on job duties and salary, present the candidates to the hiring manager by setting up the interviews. Present the offer to the candidate if selected and then sometimes run the background check.

Please remember one thing that recruiters are your best friend in this process. They would try their best to get you the interview. Think of them as your agent who is working on your behalf but at the same time if you do not qualify their initial criteria then they will drop you because they don't want to tarnish their reputation with the hiring manager.

Now who are the HR people? The HR is the human resources department in a companies who deals directly with the candidates. They usually do not search for you online. They will only contact if you have applied for a job on the company website or through one of the job search websites. If you do not qualify for the job requirement then simply an automated email will be sent out to you indicating that they are moving on with other candidates. If they like your resume then they will contact you directly through the phone or email you have specified on your resume.

- **Job description**

**ENTER →** What will be the second step when a recruiter calls you?

The recruiter will send you the job description and calls you to go over your experience and qualification. He or she will try to match job requirement with your skills. This will be your first and initial round of selection. At this time if the recruiter feels confident that you can be the right candidate then the conversation continues to the next step

- **Full time or Consultant**

**ENTER →** The recruiter lets call him Mr. Litman. Mr Litman will tell you if this job is a full time or as a consultant. Full time is when you are working directly with the hiring company. A consultant is someone who works either independently on 1099 or on W2 which will be an employee of the recruiting agency. By the way the 1099 or W2 are the tax terms used in the United States. It could be different according to tax department of your country. Anyway, Mr. Litman will specify all these details up front. He will also tell you the duration of the consultancy. It can be either 6 months to a year or unspecified number of years.

- **Rate negotiation**

**ENTER →** Now the money part. Mr. Litman at this point asks you about your salary expectation or per hour rate. Tell him that you are flexible and willing to negotiate or if you are already in IT then you should have a better idea of what you ask for.

- **Updated resume with rate confirmation**

**ENTER →** When agreed to a rate then Mr. Litman will send you an email and he would ask you to acknowledge the rate confirmation. He will also ask you to send him your updated resume if there is some updates required. Often times this email will also have some questions for you to fill out. E.g. your full legal name, your address, telephone number, if you are authorized to work in the country, your last 4 digits of social security, or birth month and day. Now please don't be scared to provide that information. They are not here to steal your identification, this is how they keep track of candidates in their database.

- **Quick interview with the recruiter's manager over the phone**

**ENTER →** The next step is that Mr. Litman might ask you to talk to his manager who is also in the recruiter agency. There will be a quick 10 minutes interview with Mr. Litman's manager who will also go through your resume, your background and if you are ready to be presented.

- **Meet the recruiter in person**

ENTER → Sometimes Mr. Litman or his manager will ask you to come and meet them in person if they are in the same city as yours. This will be the perfect opportunity for you to go and meet them and start practicing your in-person interview skills. You can deny the meeting request but perhaps it might effect your chances to be presented. I would recommend that you make some time and go see them.

- **Resume submission**

ENTER → After all the conversation and formalities with Mr. Litman, your resume will be submitted to the hiring manager. It can take anywhere from 2 days to a couple of weeks for the hiring manager to get back to Mr. Litman.

- **Interview schedule**

ENTER → Once your resume is reviewed by the hiring manager and shortlisted then Mr. Litman will be notified and a request for interview will be placed. Mr. Litman will reach out to you with a few interview time slots. Pick your desired time slot and start the preparation. If the hiring manager does not like your resume then he will notify Mr. Litman. If you don't hear back from Mr. Litman for 3-4 weeks then you should assume that it's a NO. You can always call Mr. Litman for a follow up.

- **First phone interview**

ENTER → So lets be optimistic. You are scheduled for an interview. The first interview will be most likely a phone interview and it can last up to an hour. This can include technical questions as well.

- **Second in-person interview**

ENTER → A couple of days later you will get a call from Mr. Litman you will provide feedback based on your interview. If you have cleared the first interview then he will schedule the second round of interview. This will be most likely on-site in-person interview and can include additional technical interview questions. You will meet with the team so make sure you are dressed with a suit and tie and present yourself professionally.

- **Offer Letter**

NEXT → A few days go by and your phone will ring. Mr. Litman will give you the great news that you have been selected and the hiring manager is ready to offer you the job. You of course jump off the bed and ask for more details about the start date.

- **Acceptance and start date.**

NEXT → Mr. Litman will send you all the required paper work to fill out and sign the offer letter. Also at this point you might have to go through the background check which might include, education history, experience, criminal record and sometimes a drug test depending on the hiring company requirements.

When you start the first day then you will feel that your hard work has finally paid off. Congratulate yourself.

**Redhat Certifications:**

**RHCSA**

<https://www.redhat.com/en/services/training/ex200-red-hat-certified-system-administrator-rhcsa-exam>

**RHCE:**

<https://www.redhat.com/en/services/training/ex300-red-hat-certified-engineer-rhce-exam>

**RHCA:**

<https://www.redhat.com/en/services/certification/rhca>

## The In-Person Interview Tips

As you know, in order to extract an offer it will not only take your understanding of the job requirements, but also your ability to demonstrate your communication, logical and behavioral skills. Interviewing is an art form and it takes combining technical skills, interpersonal communication (soft) skills, previous responsibilities and accomplishments.

This process we have outlined will help you feel more comfortable with the phone interview, assist in gauging the client's need, and establish why you are the best possible candidate.

### **Before the Interview:**

#### **1. Be presentable and educated.**

- Dress to Impress –Professional Business Attire always
  - Don't cut corners when it comes to ironing or dry-cleaning.
  - It's better to be over-dressed than under-dressed
- Try to avoid strong-scented perfumes and colognes.
- Review the company's website, job description and gather data
  - Check out newsletters, managers' online profiles, company's competitors, etc.
  - Consider a couple of questions that indicate to the hiring manager that you have researched the company – research indicates your interest!
- Know the name of the interviewer so that you can ask for that person at the receptionist's desk. It's embarrassing when the receptionist asks, 'Who are you here to see?' and you can't remember

#### **2. Eat before the interview, not during your conversation.**

- Do not bring anything into the interview such as food, drink, gum etc.
- Make sure that what you *do* eat beforehand does not have a strong odor.
  - You want to be remembered for your professionalism and outstanding skills, not for what you ate for lunch.
- You are welcome to bring a portfolio, pen and pad of paper to your interview.

#### **3. Don't be too early and bring a photo ID**

- Arrive 5-7 minutes early
- Arriving earlier will lead to anxiety – you will have to sit and wait for the hiring manager, and s/he may feel frustrated and rushed knowing you are waiting.
- If you arrive at the location early, wait in your car or take a walk around the block and review your resume.
- Reminder – turn off your cell phone before you enter the building, or better yet, leave it in your car.
- **You're late for the interview** - Say you're late for an interview.
  - Don't tell them about traffic, bad directions or parking problems. Just say, "I'm sorry I'm late. If I've thrown off your day, I will be glad to reschedule whenever it's convenient for you."
  - Take ownership, don't make excuses, and offer ways to make things better. Nothing ever goes perfectly, and knowing you will take responsibility and work to fix problems is impressive.

#### **5. The interview begins the moment you enter the building**

- Most people would never think of the receptionist as being an interviewer, but it's true. It's fairly common that the receptionist will report the candidate's behavior to the manager.
- Turn off your cell phone and wait patiently.

### **First Impressions:**

#### **1. Firm handshake, eye contact, and thank you.**

- When introduced to the interviewer, give him/her a firm handshake
- Be aware of “sweaty palms” and have a tissue in your pocket
- **Immediately thank them for taking the time to interview you** and demonstrate that you are excited about the opportunity.
- Let them direct you where to sit and don’t sit before they do. Walking in and sitting down immediately gives the impression that you feel like you own the place.
- Eye contact indicates you are interested and engaged in the conversation.
  - Any given eye contact should last about five seconds at a time. Any more can be intimidating; any less could be seen as insincere.
  - If there is more than one interviewer, make eye contact with the interviewer 60 percent of the time, and 40 percent should be given to the other people in the room.

#### **2. Don't look at your watch, click your pen, or touch your face repeatedly**

- Don’t have distractions during the interview. The manager should be focused on your skills, not the drumming of your fingers on the table, the ringing of your phone or your constant itch.

#### **3. Be ready and attentive.**

- Be engaged and set to go. Hit the ground running and immediately focus on the interviewer. “Work” is a verb. Make “interview” a verb too.
- Sit up, sit forward, and show the interview matters to you; kicking back says you don’t really care.
- If you brought a pad of paper or a portfolio, don’t take notes the entire conversation. Eye contact is imperative!

### **Discover the Need during the Interview (Very Important)**

#### **1. REMEMBER – the client has a need, and if they did not, you wouldn’t be sitting there for your interview. Your goal is to reconfirm/establish what the client is seeking in a candidate.**

- Ask specifics about the position in the BEGINNING OF THE INTERVIEW, and mirror your background to the specific need.
- You have to figure out what the client needs in a candidate. If it is not described to you, you have to ask.
- Make the interview into more of a conversation vs. a Q&A session.

#### **2. Use the interviewer’s name when addressing him or her; it grabs their attention.**

#### **3. Don’t interrupt or overpower the interviewer: best interviews are the ones where the manager speaks the most.**

- Everyone appreciates a leader, but when you overpower the interview, it is very irritating.

#### **4. Use “I” when stating responsibilities. Your previous colleagues and teammates are not interviewing for the job, you are, so keep it that way.**

#### **5. FOCUS**

- **Demonstrate Interest** by asking questions, regarding technology, products, methodologies etc.
- **Turn negatives into positives – always tell the client what you can do, not what you can’t do.**  
**If you are asked a question and you’ve never worked with that specific tool, but you know**

**what it is – demonstrate familiarity and/or talk about similar products you have worked with.**

- **Provide examples** of similar tools & technologies you have used and where (“I’ve never worked with Waterfall methodology, but I have worked with SDLC and PMBOK.”)
  - **Show** that you have some knowledge of the product or tool and interest in learning more about it. “I have never worked with QTP, but I know it is an automated testing tool.”
  - **Don’t complain.** Most people know not to complain about their present employer, but any complaint is a downer. Even gentle whining is a bummer. Don’t complain about anything, no matter how justified. Negatives always stand out.
6. **IDLE THE CONVERSATION: Once or twice in the middle of the interview, ask the client “are you satisfied with that answer?” or “do you want me to explain something further?”**
- How do you know you provided the right answer if you don’t ask? Never assume.
  - Remember the difference between “open ended” and “close ended” questions and answers.
  - Be prepared to stand behind your answer – whatever you provide for an answer is likely to lead to another question to see if you can back up your statement

**End of Interview:**

1. The client will always indicate to you that they want the interview to end. More than likely, the interviewer will say to you “that completes our interview for today, do you have any questions or comments?”
  - At this point, if you have a burning question that is specific to your conversation, go ahead and ask it.
  - **HOWEVER, you should NOT ask about the project and what they are looking for in a candidate. IT IS TOO LATE.**
2. **What NOT to do**
  - Do not ask broad questions which should have already been answered i.e. “what are the project details.” If this question hasn’t been answered already, it would be tough to get it answered in the 30 seconds remaining in the interview.
  - Do not ask for feedback i.e. “how did I do?” backs the manager into a corner.
  - Do not ask for the next steps i.e. “what is the next step in the hiring process?”
  - Do not make assumptive statements i.e. “look forward to working with you.”
3. **What TO do...Do not wait for the hiring manager to ask you these questions you should just come out and state the below quickly**
  - IMMEDIATELY thank them again for taking the time to speak with you
  - Demonstrate interest in the position
  - Let them know that IF you are selected you can be available....”provide specific date, day, two weeks etc.” If you are working now, full time, let them know it’s a two week notice.
  - Let the hiring manager know you are aware of the location and can work on site (if the position requires it).

**Please note:**

1. **Give your recruiter a call as soon as possible. Your feedback will allow staff to go back to the hiring manager for next steps or offer extraction.**

**Top Interview Questions:**

**Prep for the Top 10 Interview Questions**

As the saying goes, "If you fail to plan, you plan to fail". So here is a valuable insight into the world of interview questions and the techniques best used to answer them.

There are some questions that are asked frequently in interviews and you should prepare your answers beforehand. The key things to remember when responding to interview questions are to keep your answers relevant, brief and to the point. If you are faced with a difficult question, make sure you stay calm, don't get defensive, and take a moment to think about your response before you answer.

Remember, these responses are only suggestions. Try to personalize your response as much as possible.

**Question: Tell me about yourself.**

**Answer:** Identify some of your main attributes and memorize them. Describe your qualifications, career history and range of skills, emphasizing those skills relevant to the job on offer. Don't drone on for several minutes; keep it concise and to the point.

**Q: What have your achievements been to date?**

**A:** Select an achievement that is work-related and fairly recent. Identify the skills you used in the achievement and quantify the benefit it had to the company. For example, 'my greatest achievement has been to design and implement a new sales ledger system, bringing it in ahead of time and improving our debtors' position significantly, saving the company \$50,000 per month in interest'.

**Q: Are you happy with your career to date?**

**A:** This question is really about your self-esteem, confidence and career aspirations. The answer must be 'yes', followed by a brief explanation as to what it is about your career so far that's made you happy. If you have hit a career plateau, or you feel you are moving too slowly, then you must qualify your answer.

**Q: What is the most difficult situation you have had to face and how did you tackle it?**

**A:** The purpose of this question is to find out what your definition of difficult is and whether you can show a logical approach to problem solving. In order to show yourself in a positive light, select a difficult work situation which was not caused by you and which can be quickly explained in a few sentences. Explain how you defined the problem, what the options were, why you selected the one you did and what the outcome was. Always end on a positive note.

The need a specific **S**ituation, **T**ask at hand, **A**ction you took and the **R**esult (STAR method). This will demonstrate to them what you will likely do in other situations. Answer should be concise and to the point...no need for elaborate stories.

**Q: What do you like about your present job?**

**A:** This is a straightforward question. All you have to do is make sure that your 'likes' correspond to the skills etc. required in the job on offer. Be enthusiastic; describe your job as interesting and diverse but do not overdo it - after all, you are looking to leave.

**Q: What do you dislike about your present job?**

**A:** Be cautious with this answer. Do not be too specific as you may draw attention to weaknesses that will leave you open to further problems. One approach is to choose a characteristic of your present company, such as its size or slow decision-making processes etc. Give your answer with the air of someone who takes problems and frustrations in your stride as part of the job.

**Q: What are your strengths?**

**A:** This is one question that you know you are going to get so there is no excuse for being unprepared. Concentrate on discussing your main strengths. List three or four proficiencies e.g. your ability to learn quickly, determination to succeed, positive attitude, your ability to relate to people and achieve a common goal. You may be asked to give examples of the above so be prepared.

**Q: What is your greatest weakness?**

**A:** Do not say you have none - this will lead to further problems. You have two options - use a professed weakness such as a lack of experience (not ability) on your part in an area that is not vital for the job. The second option is to describe a personal or professional weakness that could also be considered to be a strength, and the steps you have taken to combat it. An example would be, 'I know my team think I'm too demanding at times - I tend to drive them pretty hard but I'm getting much better at using the carrot and not the stick'.

**Q: Why do you want to leave your current employer?**

**A:** State how you are looking for a new challenge, more responsibility, experience and a change of environment. Do not be negative in your reasons for leaving. It is rarely appropriate to cite salary as your primary motivator.

**Q: Why have you applied for this particular job?**

**A:** The employer is looking for evidence that the job suits you, fits in with your general aptitudes, coincides with your long-term goals and involves doing things you enjoy. Make sure you have a good understanding of the role and the organization, and describe the attributes of the organization that interest you most.

**Other questions to consider:**

- How does your job fit in to your department and company?
- What do you enjoy about this industry?
- Give an example of when you have worked under pressure.
- What kinds of people do you like working with?
- Give me an example of when your work was criticized.
- Give me an example of when you have felt anger at work.
- How did you cope and did you still perform a good job?
- What kind of people do you find it difficult to work with?
- Give me an example of when you have had to face a conflict of interest at work.
- Tell me about the last time you disagreed with your boss.
- Give me an example of when you haven't got on with others.
- Do you prefer to work alone or in a group? Why?
- This organization is very different to your current employer - how do you think you are going to fit in?
- What are you looking for in a company?
- How do you measure your own performance?
- What kind of pressures have you encountered at work?
- Are you a self-starter? Give me examples to demonstrate this?
- What changes in the workplace have caused you difficulty and why?
- How do you feel about working long hours and/or weekends?
- Give me an example of when you have been out of your depth.
- What have you failed to achieve to date?
- What can you bring to this organization?

**Linux 200+ Technical Interview Questions:**

*In some questions I am encouraging to search online because it will help you practice for searching more complicated questions in the future  
(Good Luck)*

1. When you login you get “\$” prompt, what is the prompt for root?  
**#**
2. Explain the difference between grep and egrep?  
**Search online**
3. What is the port # for DNS, NTP and NFS?  
**53,123 and 111/2049**
4. What is the configuration file name of DNS and where is it located?  
**/etc/named.conf**
5. How many new directories will be created after running the following command  
mkdir {a..c}{1..3}  
**9**
6. Your PC is configured with a DNS server address but not the default gateway. Can the PC access internet?  
**No**
7. What is the difference between IP and Gateway?  
**Search online**
8. Can you assign one static IP to 2 computers, if not then why?  
**No because it will create IP conflict**
9. How to change IPs address to static?  
**ifconfig x.x.x.x**
10. You are trying to ping a server by hostname and you get an error message, “ping: unknown host ...”. What could be the reason and how to solve the problem so you can ping it by hostname?  
**Check for /etc/hosts or DNS to see if it has hostname to IP entry**
11. Explain the difference between relative and absolute path?  
**Absolute path starts from / where relative path is your current directory**
12. List 3 different methods of adding user?  
**Search online**
13. What is the command to change file/directory ownership and group?  
**chown and chgrp**
14. List any 3 type of filesystem?  
**ext4,NTFS and FAT**

15. When you login you get a message on the screen. What is the name of that file and where is it located?  
**/etc/motd**
16. What is /bin directory used for?  
**Search online**
17. What are the different types of DNS Server  
**Master and secondary**
18. How to change a user password?  
**passwd username**
19. What is the version of Redhat Linux you have experience with?  
**7.4**
20. List any 4 linux distributions?  
**Redhat, CentOS, Ubuntu and SUSE**
21. How to logoff from the system?  
**exit**
22. Give any 3 examples of operating systems?  
**Windows, Linux and MAC**
23. How to create a directory?  
**mkdir**
24. Where are the zone files located for DNS service?  
**/var/named/zonfiles**
25. How to check kernel version?  
**uname -a**
26. Which directory has all the configuration files?  
**/etc**
27. How to become root user from a regular user?  
**su -**
28. How many mega bytes in 1 giga bytes?  
**Search online**
29. What is the purpose of having different network ports?  
**So the communication of each application goes through a dedicated port**
30. How to display first column of a file?  
**cat filename | awk '{print \$1}'**
31. What is the name of DNS rpm package?  
**bind**
32. What is the difference between nslookup and dig commands?  
**Search online**
33. How to check your user id and group id?

- id**
34. How to check a file's permission?  
**ls -l**
35. What is the difference between "kill" and "kill -9" command?  
**Search online**
36. What is subnet?  
**Search online**
37. You are troubleshooting an issue with Redhat support and they have asked you to send the contents of /etc directory. How and which method you will use to transfer the contents?  
**tar (compress) the entire /etc directory and ftp**
38. What is root home directory?  
**/root**
39. What is rsyslogd deamon and its purpose?  
**Search online**
40. Your company has terminated a server administrator. What is first thing as an administrator you should do to enhance the security?  
**Change root password**
41. How to check the computer name or host name in Linux?  
**hostname**
42. Which permission allows a user to run an executable with the permissions of the owner of that file?  
**First 3 bits should have x**
43. What is the command to untar a tarred file?  
**untar**
44. What is /proc directory used for?  
**Search online**
45. What is the purpose of nsswitch.conf file  
**It tells the system where to go to resolve hostnames**
46. List 3 basic commands to navigate the filesystem?  
**cd, pwd and ls**
47. Which service/daemon should be running on the server that allows you to connect remotely?  
**sshd**
48. What is the purpose of firewall?  
**Search online**
49. List any 3 IT components?  
**Hardware, OS and Applications**
50. Which directory has all the commands we use, e.g. ls, cd etc.?  
**/usr/bin or /bin**

51. What is the difference between memory, virtual memory and cache?  
**Search online**
52. Which of the following is correct?  
a. **Hardware → Operating System → Users**  
b. Operating System → Users → Hardware  
c. Database → Hardware → Users
53. Which of the following is a communication command?  
 grep  
 **mail**  
 touch  
 cd
54. How to rename a file or directory?  
**mv**
55. How to change a hostname in Linux?  
**Search online**
56. How to check network interfaces in Linux?  
**ifconfig**
57. Why is "tail -f logfilename" command used most often and what does it do?  
**It will output all incoming logs in real time**
58. What type of hardware have you worked on?  
**You should get yourself familiar with Dell, HP and UCS hardware by going online and check the vendor websites**
59. How to sort a file in reverse order?  
**cat filename | sort -r**
60. What is the name of operating system that runs Unix?  
**Solaris, HP-UX etc.**
61. List all byte sizes from smallest to largest?  
**Search online**
62. How to check the total number of partition in Linux?  
**fdisk -l**
63. How to access a linux system from a linux system?  
**ssh**
64. Explain the procedure of bonding 2 NICs or interfaces together?  
**Search online**
65. What is the exact command syntax to list the 5<sup>th</sup> column of a file and cut the first 3 letters?  
**cat filename | awk '{print \$5}' | cut -c1-3**
66. What is /etc/hosts file used for?  
**To resolve hostnames with IP address**

67. List any 3 options of 'df' command and what they are used for?  
**Search online**
68. What is the command to change file/directory permissions?  
**chmod**
69. What is the purpose of pipe (|)?  
**To combine multiple commands**
70. What is /etc directory used for?  
**For configuration files**
71. Which command is used to list files in a directory?  
**ls -l**
72. There is a command which gives you information about other commands, please explain that command and what is it used for?  
**man**
73. How to delete a file and a directory?  
**rm filename and rmdir dirname**
74. What is the difference between "tail" and "tail -10"?  
**None**
75. List 4 commands to display or read a file contents?  
**cat, more, less, vi**
76. Which command is used to read the top 5 lines of a file?  
**head -5 filename**
77. What are the different commands or methods to write to a file?  
**echo > filename and vi filename**
78. What is swap space and how to check swap space?  
**Search online**
79. What is inode and how to find an inode of a file?  
**Search online**
80. Which file to edit for kernel tuning?  
**Search online**
81. What is the latest version of Redhat?  
**Search online**
82. Name the command to find specific word from a file?  
**grep word filename**
83. You have scheduled a job using crontab but it does not run at the time you specified, what could be the reason and how would you troubleshoot?  
**Check your system time**  
**Check your crontab entry**  
**Check /var/log/messages**
84. How to check system hardware information?

- dmidecode**
85. How to check network interface MAC address?  
**ifconfig**
86. If I don't want others to read my file1, how to do that?  
**Remove r from the last 3 bits of file permission**
87. What is the purpose of "uniq" and "sed" command?  
**Search online**
88. Which command is used to list the contents of a directory in the most recent time and in reverse order, meaning the most updated file should be listed on the bottom?  
**ls -ltr**
89. What is the difference between tar, gzip and gunzip?  
**Search online**
90. What are the different ways to install and OS?  
**DVD, DVD iso and network boot**
91. How to view difference between two files?  
**diff file1 and file2**
92. You noticed that one of the Linux servers has no disk space left, how would you troubleshoot that issue?  
**If running LVM then add more disk and extend LVM**  
**If not running LVM then add more disk, create a new partition and link the new partition to an existing filesystem**
93. How to check Redhat version release?  
**uname -a or /etc/redhat-release**
94. What is the difference between TCP and UDP?  
**Search online**
95. What is a zombie process?  
**Search online**
96. How do you search for a pattern/word in a file and then replace it in an entire file?  
**sed command**
97. Explain the purpose of "touch" command?  
**To create an empty file**
98. If a command hangs, how to stop it and get the prompt back?  
**Ctrl C**
99. Which command is used to count words or lines?  
**wc**
100. How to check the number of users logged in?  
**who**
101. What is the command to view the calendar of 2011?

- cal 2011**
102. Which command is used to view disk space?  
**df -h**
103. How to create a new group in Linux?  
**groupadd**
104. What is the command to send a message to everyone who is logged into the system?  
**wall**
105. Which command is used to check total number of disks?  
**fdisk -l**
106. What is an mail server record in DNS?  
**MX**
107. What does the following command line do?  
`ps -ef | awk '{print $1}' | sort | uniq`  
**List the first column of all running processes, sort them and remove duplicates**
108. You get a call that when a user goes to [www.yourwebsite.com](http://www.yourwebsite.com) it fails and gets an error, how do you troubleshoot?  
**Check for user internet**  
**Check to see if user computer has DNS for hostname lookup**  
**Check to see if the server is up that is running that website**  
**Check to see if the server's web service is running**  
**Check for DNS availability which is resolving that website**
109. List 4 different directories in /?  
**/etc, /bin, /tmp, /home**
110. What is the output of the following command:  
`$tail -10 filename | head -1`  
**It will show the first line from the last 10 lines of a file**
111. What are the different fields in /etc/passwd file?  
**Search online**
112. Which command is used to list the processes?  
**ps -ef**
113. What is the difference between "hostname" and "uname" commands?  
**Hostname will give you system name and uname will give you OS information**
114. How to check system load?  
**top and uptime command**
115. How to schedule jobs?  
**crontab and at**
116. What is the 3rd field when setting up crontab?  
**Day of the month**

117. What is the command to create a new user?  
**useradd**
118. What is the "init #" for system reboot?  
**6**
119. How to restart a service?  
**systemctl restart servicename**
120. How to shutdown a system?  
**shutdown or init 0**
121. What is "ftp" command used for?  
**To transfer files from one computer to another**
122. Explain cron job syntax? First is minute, second is..?  
**Min, house, day of the month, month, day of the week and command**
123. How to delete a package in Linux?  
**rpm -e packagename**
124. What is the file name where user password information is saved?  
**/etc/shadow**
125. Which command you would use to find the location of chmod command?  
**which chmod**
126. Which command is used to check if the other computer is online?  
**ping othercomputer**
127. Please explain about LAN, MAN and WAN?  
**Search online**
128. How to list hidden files in a directory?  
**ls -la**
129. What is the difference between telnet and ssh?  
**ssh is secure where telnet is not**
130. How to run a calculator on Linux and exit out of it?  
**bc and quit**
131. List any 4 commands to monitor system?  
**top, df -h, iostat, dmesg**
132. You are notified that your server is down, list the steps you will take to troubleshoot?  
**Check the system physically  
Login through system console  
Ping the system  
Reboot or boot if possible**
133. What is difference between static and DHCP IP?  
**Search online**
134. How to write in vi editor mode?  
**i = insert, a = insert in next space, o = insert in new line**

135. What is the difference between "crontab" and "at" jobs?  
**crontab is for repetitive jobs where at is for one time job**
136. What is vCenter server in VMWare?  
**Search online**
137. What is "dmidecode" command used for?  
**To get system information**
138. What is the difference between SAN and NAS?  
**Search online**
139. What is the location of system logs? E.g. messages  
**/var/log directory**
140. How to setup an alias and what is it used for?  
**alias aliasname="command"**  
**It is used to created short-cuts for long commands**
141. What is the purpose of "netstat" command?  
**Search online**
142. What are terminal control keys, list any 3?  
**Crtl C, D and Z**
143. Which command(s) you would run if you need to find out how many processes are running on your system?  
**ps -ef | wc -l**
144. What are the different types of shells?  
**sh, bash, ksh, csh etc.**
145. How to delete a line when in vi editor mode?  
**dd**
146. Which is the core of the operating system?  
a) Shell  
**b) Kernel**  
c) Commands  
d) Script
147. Which among the following interacts directly with system hardware?  
a) Shell  
b) Commands  
**c) Kernel**  
d) Applications
148. How to save and quit from vi editor?  
**Shift ZZ or :wq!**
149. What is the difference between a process and daemon?  
**Search online**
150. What is the process or daemon name for NTP?  
**ntpd**

151. What are a few commands you would run if your system is running slow?  
**top, iostat, df -h, netstat etc.**
152. How to install a package in Redhat Linux?  
**yum install packagename**
153. What is the difference between "ifconfig" and "ipconfig" commands?  
**ifconfig for Linux and ipconfig for Windows**
154. What is the first line written in a shell script?  
**Define shell  
e.g. #!/bin/bash**
155. Where is the network (Ethernet) file located, please provide exact directory location and file name?  
**/etc/sysconfig/network-scripts/ifcfg-nic**
156. Why do we use "last" command?  
**To see who has logged in the system whether active or logged off**
157. What is RHEL Linux stands for?  
**Search online**
158. To view your command history, which command is used and how to run a specific command?  
**history and history #**
159. What is NTP and briefly explain how does it work and where is the config files and related commands of NTP?  
**Search online**
160. How to disable firewall in Linux?  
**Search online**
161. How to configure mail server relay for sendmail service?  
**Edit /etc/mail/sendmail.mc file and add SMART\_HOST entry**
162. Where is samba log file located?  
**/var/log/samba**
163. What is mkfs command used for?  
**To create a new filesystem**
164. If you create a new group, which file does it get created in?  
**/etc/group**
165. Which file has DNS server information (e.g. DNS resolution)?  
**/etc/resolv.conf**
166. What are the commands you would run if you need to find out the version and build date of a package (e.g. http)?  
**rpm -qi http**

167. On the file permissions? What are the first 3 bits for and who is it for?  
**Read, write and execute. They are used for the owner of the file**
168. How to create a soft link?  
**ln -s**
169. How to write a script to delete messages in a log file older than 30 days automatically?  
**Search online**
170. How to quit out of "man" command?  
**q**
171. Which command is used to partition disk in Linux?  
**fdisk**
172. What is the difference between "shutdown" and "halt" command?  
**Search online**
173. What is the exact syntax of mounting NFS share on a client and also how to un-mount?  
**Search online**
174. What experience do you have with scripting, explain?  
**if-the, do-while, case, for loop scripts**
175. How to get information on all the packages installed on the system?  
**rpm -qa**
176. Explain VMWare?  
**Search online**
177. You are tasked to examine a log file in order to find out why a particular application keep crashing. Log file is very lengthy, which command can you use to simplify the log search using a search string?  
**grep for error, warning, failure etc. in /var/log/messages file**
178. What is /etc/fstab file and explain each column of this file?  
**Search online**
179. What the latest version of Windows server?  
**Search online**
180. What is the exact command to list only the first 2 lines of history output?  
**history | head -2**
181. How to upgrade Linux from 7.3 to 7.4?  
**yum install update**
182. How to tell which shell you are in or running?  
**\$0**

183. You have tried to "cd" into a directory but you have been denied. You are not the owner of that directory, what permissions do you need and where?

- - - - - r - x

184. What is CNAME record in DNS?

**Entry for hostname to hostname**

185. What is the name of VMWare operating system?

**ESXi**

186. What is the client name used to connect to ESXi or vCenter server?

**vSphere client**

187. You get a call from a user saying that I cannot write to a file because it says, permission denied. The file is owned by that user, how do you troubleshoot?

**Give write permission on the first 3 bits**

188. What is the latest version of VMWare?

**Search online**

189. What is the name of firewall daemon in Linux?

**firewalld**

190. Which command syntax you can use to list only the 20<sup>th</sup> line of a file?

**Search online**

191. What is the difference between run level 3 and 5?

**3 = Boot system with networking, 5 = boot system with networking and GUI**

192. List a few commands that are used in troubleshooting network related issue?

**netstat, tcpdump etc.**

193. What is the difference between domain and nameserver?

**Search online**

194. You open up a file and it has 3000 lines and it scrolled up really fast, which command you will use to view it one page at a time?

**more or less**

195. How to start a new shell. E.g. start a new ksh shell?

**Simply type ksh, or bash**

196. How to kill a process?

**kill processID**

197. How to check scheduled jobs?

**crontab -l**

198. How to check system memory and CPU usage?  
**free and top**
199. Which utility could you use to repair the corrupted file system?  
**fsck**
200. What is the command to make a service start at boot?  
**systemctl enable servicename**
201. How to combine 2 files into 1? E.g. you 3 lines in file "A" and 5 lines in file "B", which command syntax to use that will combine into one file of  $3+5 = 8$  lines  
**cat fileA >> fileB**
202. What is echo command used for?  
**To output to a screen**
203. What does the following command do?  
echo This year the summer will be great > file1  
**It will create a new file "file1" with the content as "This year the summer will be great"**
204. Which file to modify to allow users to run root commands  
**/etc/sudoers**
205. You need to modify httpd.conf file but you cannot find it, Which command line tool you can use to find file?  
**find / -name "httpd.conf"**
206. Your system crashed and being restarted, but a message appears, indicating that the operating system cannot be found. What is the most likely cause of the problem?  
**The /boot file is most likely corrupted**

# **WELCOME TO: COURSE RECAP**

**LETS SUMMARIZE**

# **COMMANDS WE HAVE LEARNED**

160+ Commands

**CONGRATULATIONS  
YOU DID IT**

**REMEMBER HARD WORK  
ALWAYS PAYS OFF**

Commands Recap		
#	Commands	Description
1	alias	To create a short name of a long command
2	arch	Show system architecture (32 or 64bit)
3	at	Schedule ad-hoc jobs
4	awk	Shows the output by field
5	bash	Shell
6	bc	Calculator
7	bg	Run a process in the background
8	cal	Calender
9	case	Script to provide options
10	cat	Read a file
11	cd	Change directory
12	chage	Change a users attributes (e.g. password expiration etc.)
13	chgrp	Change a file group ownership
14	chmod	Change file permissions
15	chown	Change a file ownership
16	chronyc	Command for Chronyd (newer version of NTP)
17	clear	Clear the screen
18	cmp	Compare to files byte by byte
19	command --help	Quick help on a command
20	cp	Copy files or directories
21	createrepo	Create local repository
22	crontab	Schedule jobs
23	curl	Transfer data or get destination server status
24	cut	Cut characters or fields
25	date	Display date and time
26	dd	Convert or copy a file
27	df	File system disk space usage
28	diff	Compare files line by line
29	dig	DNS lookup utility

30	dmesg	Print the kernel ring buffer (system messages)
31	dmidecode	Print system hardware information
32	do	Scripting command works inconjunction with for or while loop
33	du	File space usage
34	echo	Display input on the screen
35	else	Scripting command works inconjunction with for loop
36	ethtool	Print NIC information
37	exit	Exit the terminal
38	export	export NFS filesystem
39	fdisk	Display disk information
40	fg	Bring a process in foreground
41	fi	Scripting command works inconjunction with if
42	find	Find files and directories
43	firewall-cmd	Command line for firewall
44	firewall-config	GUI for firewall
45	for	Scripting command - For loop
46	free	Show memory and swap information
47	fsck	Repair files system
48	ftp	Command to transfer files and directories
49	grep	Used for search of a keyword
50	groupadd	Create a new group
51	groupdel	Delete a group
52	gunzip	Compress a file
53	gzip	Compress a file
54	halt	Shutdown the system immediately
55	head	List first lines of a file
56	history	Shows history of all the commands
57	hostname	Shows hostname of machine
58	hostnamectl	Utility or a command to manage hostname
59	id	Print your user information
60	if	Scripting command works inconjunction with fi

61	ifconfig	Print system network information
62	ifdown	Bring down system network interface
63	ifup	Bring up system network interface
64	init	Bring system in different level
65	iostat	Input/Output status
66	ip	Replacement of ifconfig command
67	iptables	Firewall utility command
68	kill	Kill a process by process ID
69	last	Shows listing of last logged in users
70	less	Opposite of more
71	ln	Create link
72	locate	Find files and directories
73	ls	List files and directories
74	lvcreate	Create logical volume (LVM)
75	lvdisplay	Display local volume (LVM)
76	mail/sendmail	Command to send mail
77	man	Display manual for each command
78	mkdir	Create a new directory
79	mkfs.xfs	Create an XFS filesystem
80	mkswap	Make swap
81	modprobe	Program to add and remove modules (NIC Bonding)
82	more	Display a file one page at a time
83	mount	Mount a filesystem
84	mv	Move a file or directory
85	netstat	Shows network status
86	nice	Prioritize system processes
87	nohup	Command to avoid interruption upon terminal exit
88	nslookup	DNS lookup utility
89	ntp	NTP utility
90	passwd	Change user password
91	ping	Check the status of a remote system

92	pipes	Pipe is used to change the output a command
93	pkill	Kill a process by process name
94	ps	List all running processes
95	pvcreate	Create physical volume (LVM)
96	pvdisplay	Display attributes of a physical volume (LVM)
97	pvs	Report information about physical volumes (LVM)
98	pwd	Print working directory or existing directory
99	read	Command used in a script to read the input from user
100	reboot	Reboot a system
101	rm	Remove a file or directory
102	rmdir	Make a directory
103	rpm	Utility or manage system packages
104	rsync	File copying tool (Most used to copy to remote hosts)
105	scp	File copying tool (Most used to copy to remote hosts)
106	script	Record terminal activity
107	sed	Substitute text
108	sestatus	Status of SELINUX
109	sh	Shell
110	shutdown	Shutdown the system
111	sort	Sort the output
112	sosreport	Collect and package diagnostic and support data
113	split	Split files
114	ssh	Protocol command to connect to remote host
115	stat	Display file or file system status (SELINUX)
116	su -	Switch user
117	sudo	Run a command as another user (mostly root)
118	swapoff	Turn swap OFF
119	swapon	Turn swap ON
120	systemctl	Command to stop start restart a service
121	tail	Display last lines of a file
122	tar	Put multilple files in a container

123	tcpdump	Show every in and out traffic of a system
124	tee	read and output to a file and screen
125	telnet	Command to connect remote host (Insecure)
126	then	Scripting command works inconjunction with for or while loop
127	top	Print system resources
128	touch	Create a new empty file
129	traceroute	Trace network traffic
130	umount	Un mount a filesystem
131	uname	Print system information
132	uniq	Remove all duplicates
133	uptime	Shows system uptime and load information
134	useradd	Create a new user
135	userdel	Delete a user
136	usermod	Modify a user attributes
137	users	Print usernames of users currently logged in
138	vgcreate	Create a volume group (LVM)
139	vgdisplay	Display a volume group (LVM)
140	vi	vi Editor
141	w	Show who is logged on and what they are doing
142	wall	Send a message to everybody's terminal
143	wc	Count words, character, lines etc
144	wget	Network downloader program
145	whatis	Short description of a command
146	which	Shows the full path of (shell) commands
147	while	Scripting command works inconjunction with do loop
148	who	Show who is logged on
149	whoami	Print your user ID information
150	write	Send a message to another user
151	xfs_growfs	Increase the size of an XFS filesystem
152	xfs_repair	Repair XFS file system
153	yum	Downloads   installs   updates packages



## How to Open an Image File in Linux GUI

- Become root
- yum install ImageMagick -y
- display image-file

# RED HAT ENTERPRISE LINUX 5, 6, AND 7

## Common administrative commands

	RHEL5	RHEL6	RHEL7	
<b>SYSTEM BASICS</b>				
View subscription information	/etc/sysconfig/rhn/systemid	/etc/sysconfig/rhn/systemid subscription-manager identity	subscription-manager identity	
Configure subscription	rhn_register	rhn_register rhnreg_ks subscription-manager	subscription-manager <sup>1</sup> rhn_register <sup>2</sup>	
View system profile	sosreport dmidecode hwbrowser		sosreport dmidecode lshw	
View RHEL version information			/etc/redhat-release	
	1 subscription-manager is used for Satellite 6, Satellite 5.6 with SAM and newer, and Red Hat's CDN. 2 RHN tools are deprecated on Red Hat Enterprise Linux 7. rhn_register should be used for Satellite server 5.6 and newer only. For details, see: <a href="#">Satellite 5.6 unable to register RHEL 7 client system due to rhn-setup package not included in Minimal Installation</a>			
<b>BASIC CONFIGURATION</b>				
Graphical configuration tools		system-config-*	gnome-control-center	
Configure network		system-config-network	nmcli nm-tui nm-connection-editor	
Configure system language		system-config-language	localectl	
Configure time and date		system-config-date	timedatectl date	
Synchronize time and date	ntpdate /etc/ntp.conf		timedatectl /etc/chrony.conf	
Configure keyboard		system-config-keyboard	localectl	
Text-based configuration tools		system-config-* tui		
Configure printer		system-config-printer		
Configure samba		smbclient /etc/samba/smb.conf smbpasswd		
Configure SSH		/etc/ssh/ssh_config /etc/ssh/sshd_config -/.ssh/config ssh-keygen		
<b>JOBs AND SERVICES</b>				
Configure logging	/etc/syslog.conf	/etc/rsyslog.conf	/etc/rsyslog.conf /etc/rsyslog.d/*.*.conf /var/log/journal systemd-journal.service	
List all services		chkconfig --list ls /etc/init.d/	systemctl -at service ls /etc/systemd/system/*.service ls /usr/lib/systemd/system/*.service	
List running services		service --status-all	systemctl -t service --state=active	
Start/stop service		service name start service name stop	systemctl start name.service systemctl stop name.service	
Enable/disable service		chkconfig name on chkconfig name off	systemctl enable name.service systemctl disable name.service	
View service status		service name status	systemctl status name.service	
Check if service is enabled		chkconfig name	systemctl is-enabled name	
Create new service file or modify configuration		chkconfig --add	systemctl daemon-reload	
View run level/target		runlevel who -r	systemctl get-default who -r	
Change run level/target		/etc/inittab init run_level	systemctl isolate name.target systemctl set-default	
View logs		/var/log	/var/log journalctl	
Configure system audit		add audit=1 to kernel cmdline audictl /etc/audit/auditd.conf /etc/audit/audit.rules authconfig /etc/pam.d/system-auth pam_tty_audit kernel module		
View audit output		aureport /var/log/faillog		
Schedule tasks		cron at		
Configure batch tasks		batch		
Find file by name		locate		
Find file by characteristic		find		
Create archive		tar cpio zip gzip bzip2		
<b>KERNEL, BOOT, AND HARDWARE</b>				
Single user/rescue mode	append 1 or s or init=/bin/bash to kernel cmdline		append rd.break or init=/bin/bash to kernel cmdline	
Shut down system	shutdown		systemctl shutdown	
Power off system	poweroff		systemctl poweroff	
Halt system	halt		systemctl halt	
Reboot system	reboot		systemctl reboot	
Configure default run level/target	/etc/inittab		systemctl set-default	
Configure GRUB bootloader	/boot/grub/grub.conf		/etc/default/grub grub2-mkconfig grub-set-default	
View hardware configured	hwbrowser	lshw		
Configure kernel module		modprobe		
Configure hardware device		udev		
View kernel parameters		sysctl -a cat /proc/cmdline		
Load kernel module		modprobe		
Remove kernel module		modprobe -r		
View kernel version		rpm -q kernel uname -r		
<b>SOFTWARE MANAGEMENT</b>				
Install software		yum install yum groupinstall	yum install yum group install	
View software info		yum info yum groupinfo	yum info yum group info	
Update software		yum update		
Upgrade software		yum upgrade		
Configure software repository		/etc/yum.repos.d/*.repo		
Find file in package		rpm -qf filename		
View software version		rpm -q packagename		
View installed software		rpm -qa		
<b>SECURITY AND IDENTITY</b>				
Configure system security		/etc/selinux/config chcon restorecon semanage setsebool system-config-selinux		
Report on system security		sealert		
LDAP, SSSD, Kerberos		authconfig authconfig-tui authconfig-gtk		
Network users		getent		
<b>FILE SYSTEMS, VOLUMES, AND DISKS</b>				
Default file system		ext3	ext4	xfs
Defragment disk space		copy data to new file system fsck (look for 'non-contiguous inodes')	copy data to new file system fsck (look for 'non-contiguous inodes') xfs_fsr	
Create/modify disk partitions		fdisk parted	fdisk gdisk parted ssm create	
Format disk partition		mkfs.filesystem_type (ext4, xfs) mkswap	mkfs.filesystem_type (ext4, xfs) mkswap ssm create	
Mount storage		mount /etc/fstab	mount /etc/fstab ssm mount	
Create physical volume		pvcreate	pvcreate ssm create (if backend is lvm)	
Create volume group		vgcreate	vgcreate ssm create (if backend is lvm)	
Create logical volume		lvcreate	lvcreate ssm create (if backend is lvm)	
Enlarge volumes formatted with default file system		vgextend lvextend resize2fs	vgextend lvextend resize2fs	xvfs cannot currently be shrunk; copy desired data to a smaller file system.
Shrink volumes formatted with default file system		resize2fs lvreduce vgreduce		
Check/repair file system		fsck	fsck ssm check	
Configure NFS share		/etc/exports service nfs reload	/etc/exports systemctl reload nfs.service	
Mount and activate swap			/etc/fstab swapon -a	
Automatically mount at boot			/etc/fstab	
View free disk space			df	
View logical volume info			lvdisplay lvs vgdisplay vgs pvdisk pvs	
View NFS share			showmount -e mount	
Automatically mount after boot			/etc/auto.master.d/*. autofs /etc/auto.*	
Change file permissions			chmod chown chgrp umask	
Change access control list			setfacl	
<b>NETWORKING</b>				
Configure firewall		iptables and ip6tables /etc/sysconfig/iptables	iptables and ip6tables /etc/sysconfig/iptables system-config-firewall	firewall-cmd firewall-config
Configure DHCP client		/etc/dhcpd.conf /etc/dhcp6c.conf		dhcpd /etc/dhcp/dhcpd.conf /etc/sysconfig/dhcpd
Configure name resolution			/etc/hosts /etc/resolv.conf	/etc/hosts /etc/resolv.conf nmcli con mod
Configure hostname			/etc/sysconfig/network	hostnamectl /etc/hostname nmcli
View network interface info			ip addr ifconfig brctl	ip addr nmcli dev show teamdctl brctl bridge
Configure network interface			/etc/sysconfig/network-scripts/ifcfg-*	/etc/sysconfig/network-scripts/ifcfg-* nmcli con [add mod edit] nmui nm-connection-editor
View ports/sockets			ss lsof netstat	ss lsof
View routes			ip route	
Configure routes			/etc/sysconfig/network system-config-network	
<b>RESOURCE MANAGEMENT</b>				
View system usage		top ps sar iostat netstat vmstat mpstat numastat	top ps sar iostat netstat ss vmstat mpstat numastat tuna	top ps sar iostat ss vmstat mpstat numastat tuna
View disk usage		df		df iostat
Trace system calls				strace
Trace library calls				ltrace
Change process priority				nice renice
Change process run location				taskset
Kill a process				kill pkill killall