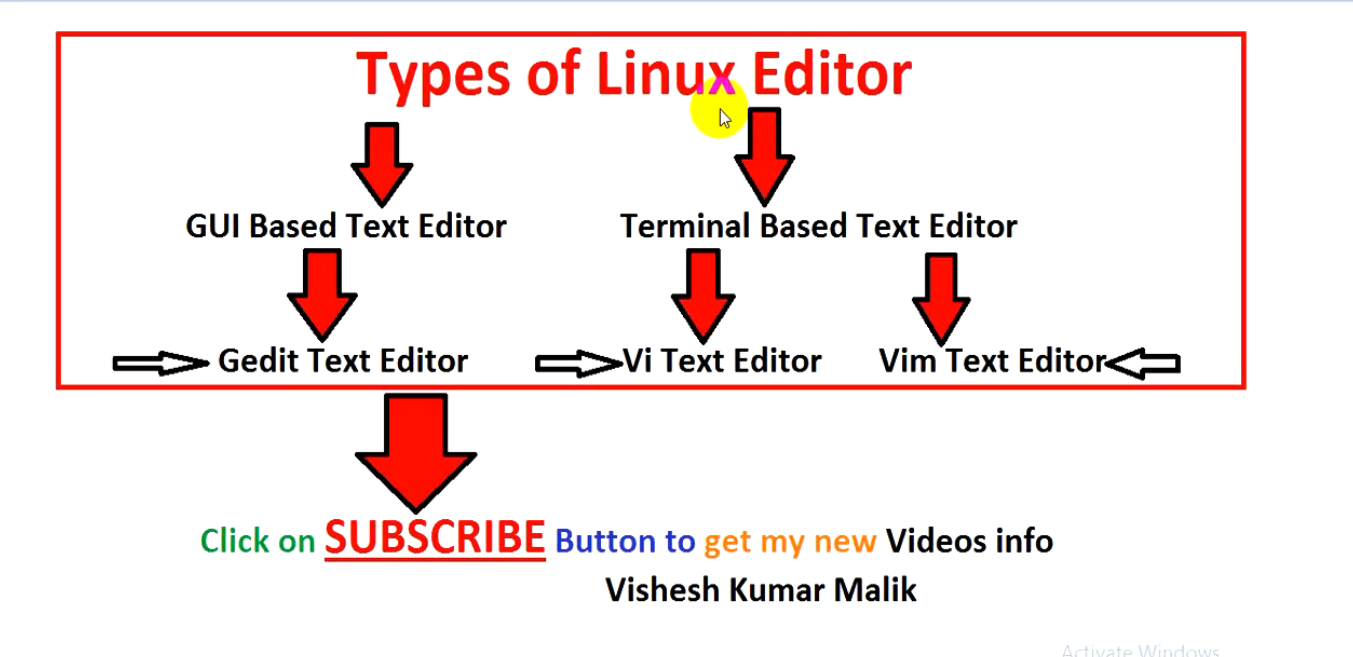
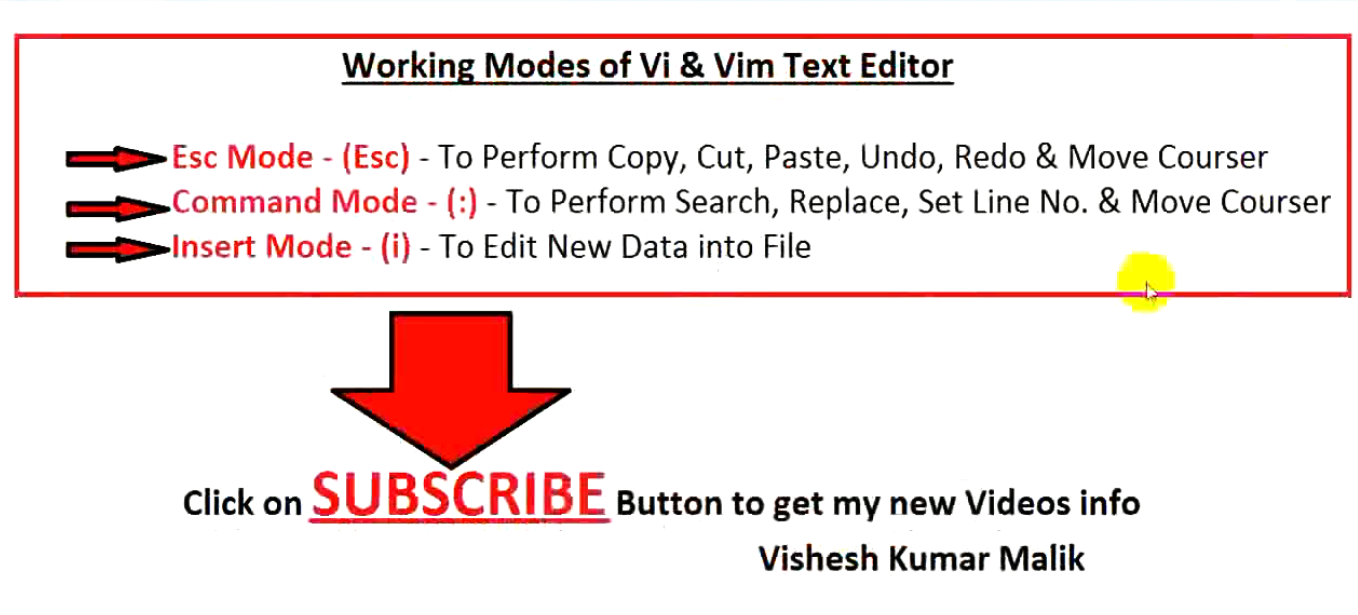
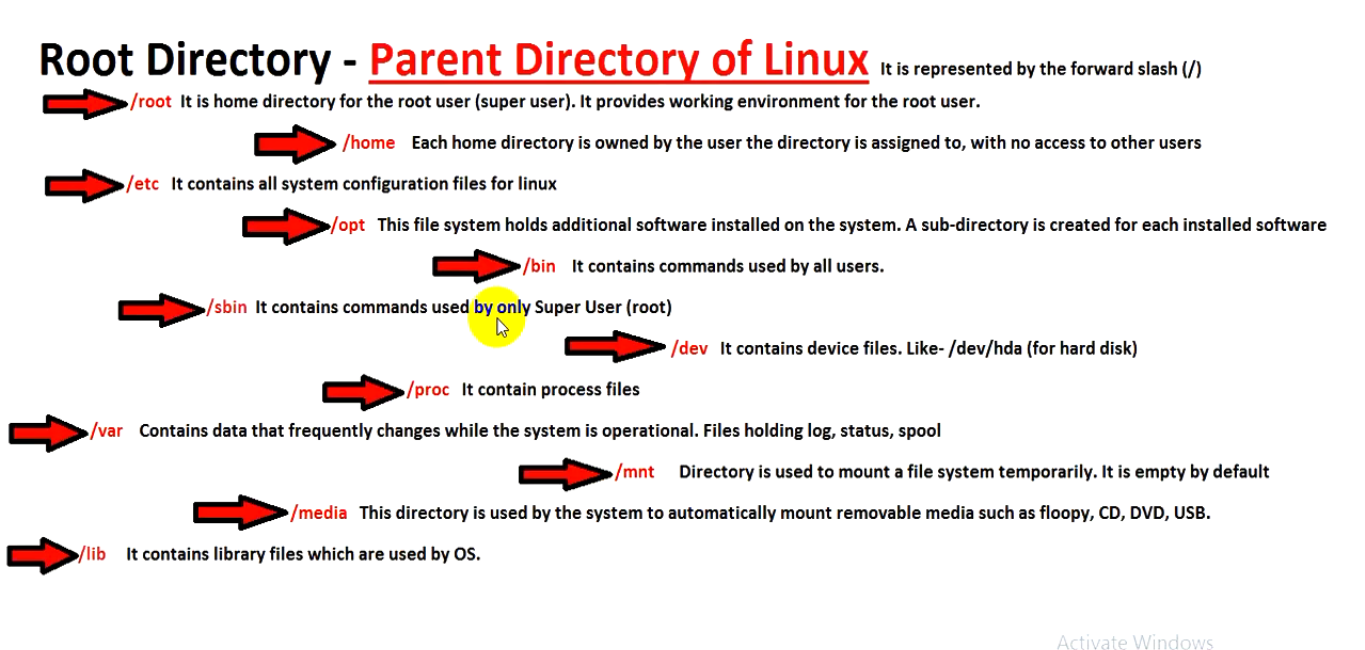
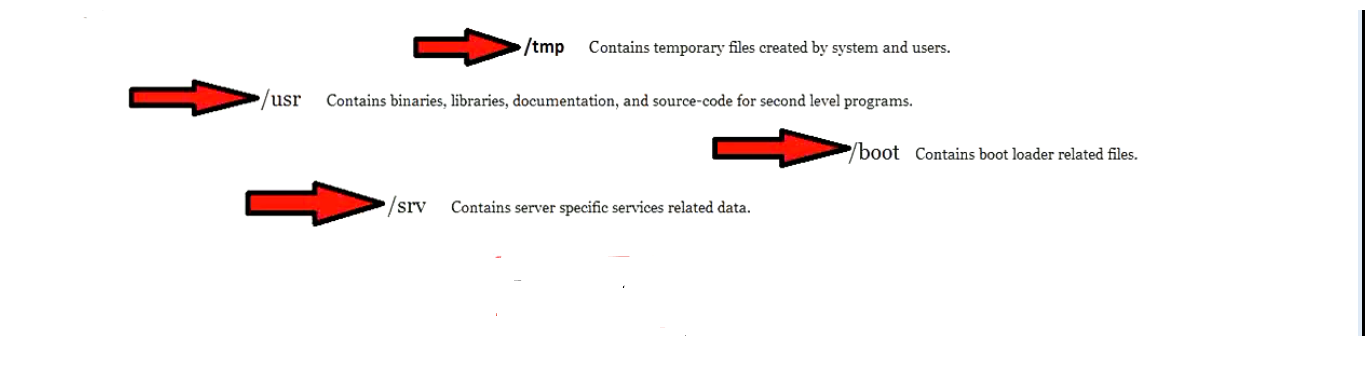
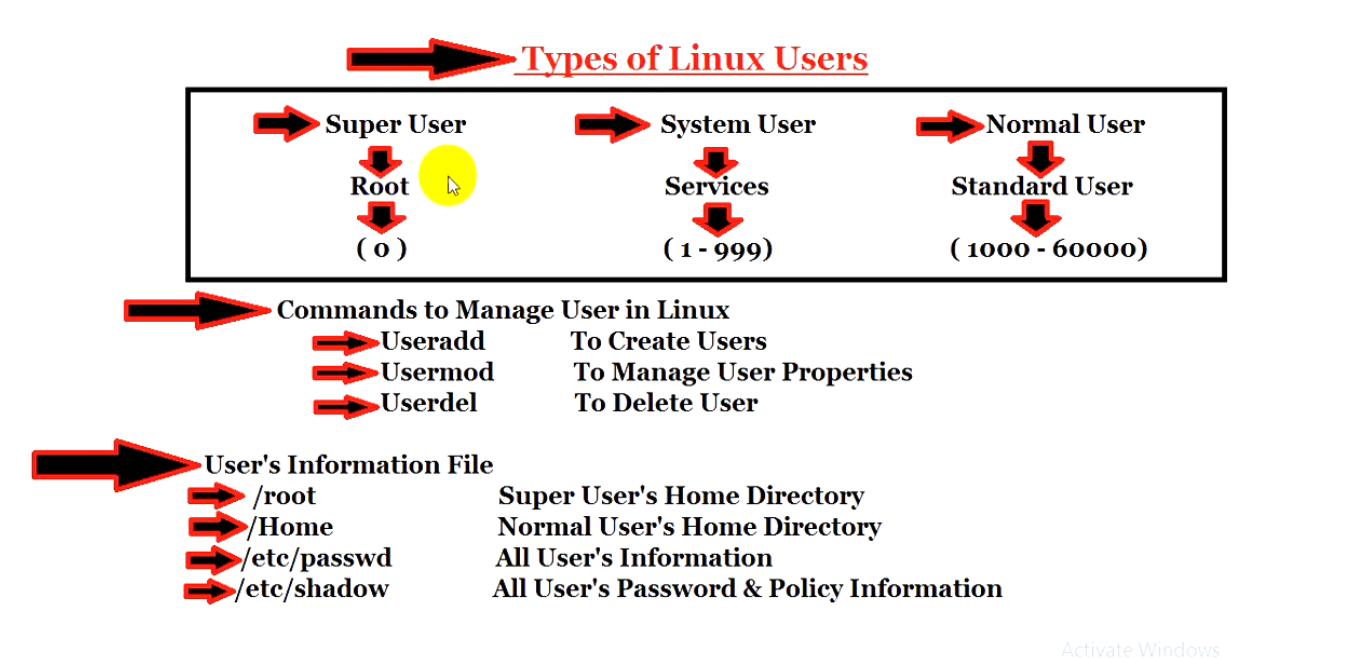
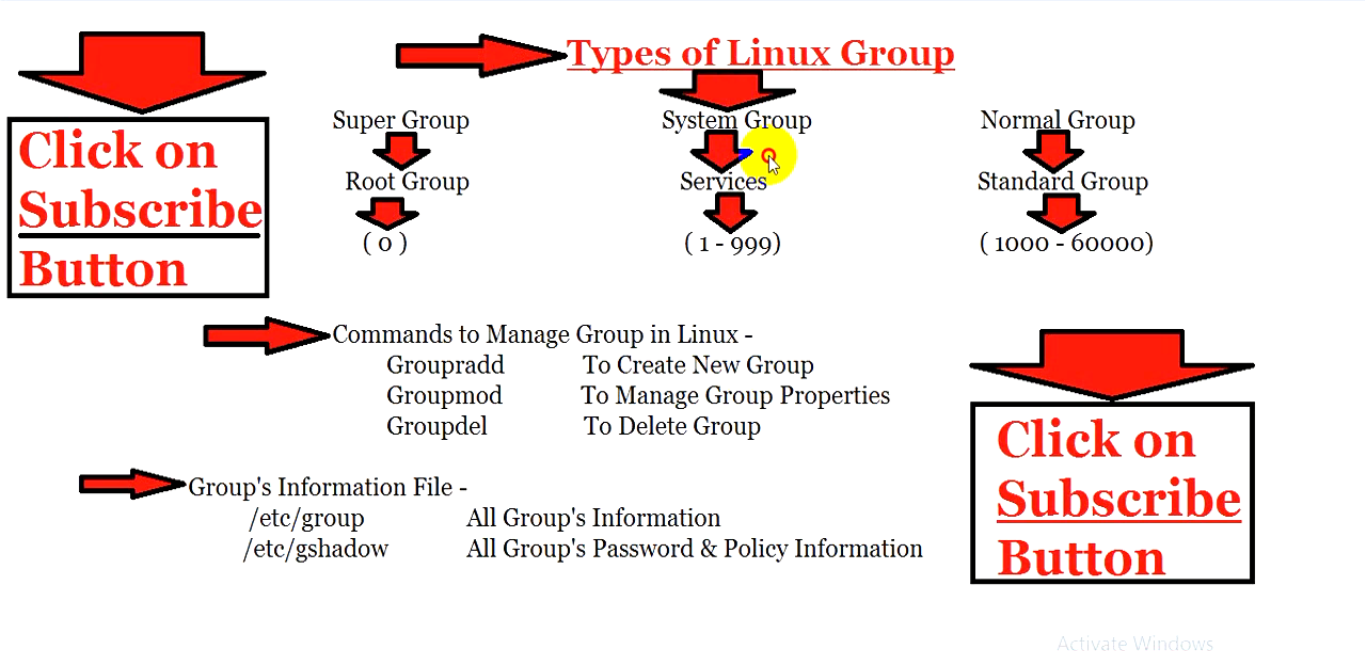


**RHEL 6 Vs RHEL 7**

|  |  |  |
| --- | --- | --- |
| **Condition** | **RHEL6** | **RHEL7** |
| **OS BOOT TIME** | **RHEL6: 40 sec** | **RHEL7: 20 sec** |
| **MAXIMUM SIZE OF SINGLE PARTITION** | **RHEL6: 50TB(EXT4)** | **RHEL7: 500TB(XFS)** |
| **BOOT LOADER** | **RHEL6: /boot/grub/grub.conf** | **RHEL7: /boot/grupb2/grub.cfg** |
| **PROCESSOR ARCHITECTURE** | **RHEL6: It support 32bit & 64bit both** | **RHEL7: It only support 64bit** |
| **HOW TO REPAIR A FILE SYSTEM IN** | **RHEL6: #fsck -y /dev/hda3** | **RHEL7: #xfs\_repair /dev/hda3** |
| **COMMAND TO MANAGE NETWORK IN RHEL6 AND RHEL7** | **RHEL6: #setup** | **RHEL7: #nmtui** |
| **HOSTNAME CONFIGURATION FILE** | **RHEL6: /etc/sysconfig/network** | **RHEL7: /etc/hostname** |
| **DEFAULT ISO IMAGE MOUNT PATH** | **RHEL6: /media** | **RHEL7: /run/media/root** |
| **FILE SYSTEM CHECK** | **RHEL6: e2fsck** | **RHEL7: xfs\_repair** |
| **RESIZE A FILE SYSTEM** | **RHEL6: #resize2fs -p /dev/vg00/lv1** | **RHEL7: #xfs\_growfs  /dev/vg00/lv1** |
| **TUNE A FILE SYSTEM** | **RHEL6: tune2fs** | **RHEL7: xfs\_admin** |
| **IPTABLES AND FIREWALL** | **RHEL6: iptables** | **RHEL7: firewalld** |
| **COMMUNICATION BETWEEN TCP AND UDP IN BACK END** | **RHEL6: netcat** | **RHEL7: ncat** |
| **INTERFACE NAME** | **RHEL6: eth0** | **RHEL7: ens198(N)** |
| **COMBINING NIC** | **RHEL6: Network Bonding** | **RHEL7: Team Driver** |
| **NSF Server Version** | **RHEL6:  NFSv2** | **RHEL7: NFSV4** |
| **DATABASE USED** | **RHEL6: Mysql** | **RHEL7: mariaDB.also support Mysql** |
| **File System.** | **RHEL6: default file system is ext4** | **RHEL7: xfs is RHEL7 default file system.** |
| **Kernel Version** | **RHEL6: default kernel version is 2.6** | **RHEL7: kernel version is 3.10** |





**Basic Permission Root & Normal user/File & Directory**

**Basic permission for File & Directory of LINUX Read/Write/Execute**

**R / W / X**

**4 / 2 / 1**

**d/l r w x = for Owner-User Owner-Group Other-User**

**Deafult Directory Permossion for Root User d/l 7 5 5 rw r r**

**Deafult File Permossion for Root User /l 6 4 4 rw r r**

**Deafult Directory Permossion for Normal User d/l 7 7 5 rwx rwx r\_x**

**Deafult File Permossion for Normal User /l 6 4 4 rw r r**

**#chmod 7 7 7 /flie-directory-name**

### Umask Permission **Root & Normal user/File & Directory**

**Umask stand for unri mask.when we created any file or directory in linux OS there default permission decide bu UMASK.**

**It’s a value that decide the default permission of any linux File and Directory.**

**#umask root (check umask value)**

**0|0 0 2**

**Default UMASK value Root User 0 2 2**

**Maximum Directory permission 7 7 7**

**Root UMAXK - 0 2 2**

**Default Directory permission 7 5 5**

**Maximum File permission 6 6 6**

**Root UMAXK - 0 2 2**

**Default File permission 6 4 4**

**Default UMASK value Normal User 0 0 2**

**Maximum Directory permission 7 7 7**

**Normal UMAXK - 0 0 2**

**Default Directory permission 7 7 5**

**Maximum File permission 6 6 6**

**Normal UMAXK - 0 2 2**

**Default File permission 6 6 4**

**For Default permission change**

**#umask 020 (set umask value)**

**After restart the OS again it will change with default UMASK vale. 022/002**

**Foe Permanent change**

**#cd**

**#l. (hidden file )**

**.bashrc**

**#vim .bashrc (set the value what you want )**

**umask 020**

**:wq (save and quit)**

## **SUID Root & Normal user/File & Directory**

## **if we want one particular command executable for all user and also future user, that time we are using SUID.SUID is defined as giving permission to a user to run a program or file with the same permission of the file owner.**

**Appling method SUID**

**#which useradd**

**/usr/sbin/useradd**

**#ll /usr/sbin/useradd (permission check)**

**#chmod u+s /usr/sbin/useradd**

**#chmod 4755 /usr/sbin/useradd**

**Remove method SUID**

**#chmod u-s /usr/sbin/useradd**

**#chmod 755 /usr/sbin/useradd**

**Sticky Bit** Permission **Root & Normal user/File & Directory**

**It’s basically used for control all other user. So we are using sticky bit because one user can’t modify / read or delete other user data.only RWX permission for Owen data.**

**#chmod o+t /directory-path (Apply\_stickbit)**

**#chmod o-t /directory-path (Remove\_stickbit)**

**#ls -ld /directory-path (check)**

**ACL** Permission **Root & Normal user/File & Directory**

## **Access control list can give access to directory or file for multiple user and group.**

**Appling method ACL**

## **# setfacl -m u: ravi : rw “user\_name”**

## **# setfacl -m g: ravi : r “user\_name”**

## **# setfacl -m o: ravi : - - - “file/dir/user name”**

## **# setfacl - - remove-all “user name” (remove all permission)**

**# getfacl /directory-name (check)**

**# getfacl “user name” (check)**

**Chattr** Permission **Root & Normal user/File & Directory**

**Provide very high level security on file or direction. It also work for normal & as well super user (root).**

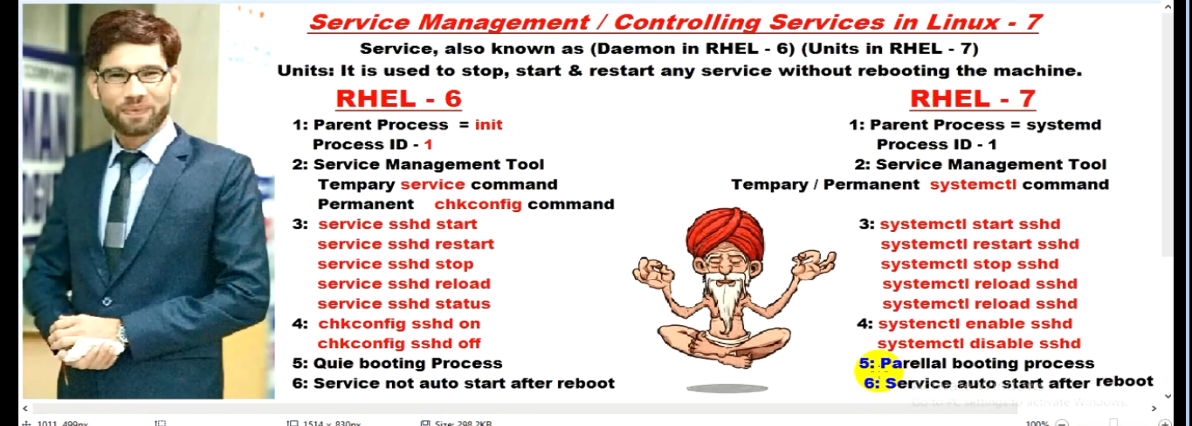
**Appling CHATTER**

**#chatter +a or +i /File-directory-name**

**Remove CHATTER**

**#chatter -a or -i /File-directory-name**

**#lsatte /File-directory-name (check)**



## 

**YUM Configuration**

**Create a directory and Copy all ISO-image file in one Directory /iso**

**#mkdir /iso**

**#cp -rvf directory-name/ to /iso/**

**#rpm -q createrepo (install repo)**

**# cd /etc/yum.repos.d/ (repo configuration directory)**

**#ls (delete all file if any file there)**

**# rm -f redhat.repo**

**#vi local.repo (create a repo configuration file and below line)**

**[local repo]**

**name=local (file name any thing)**

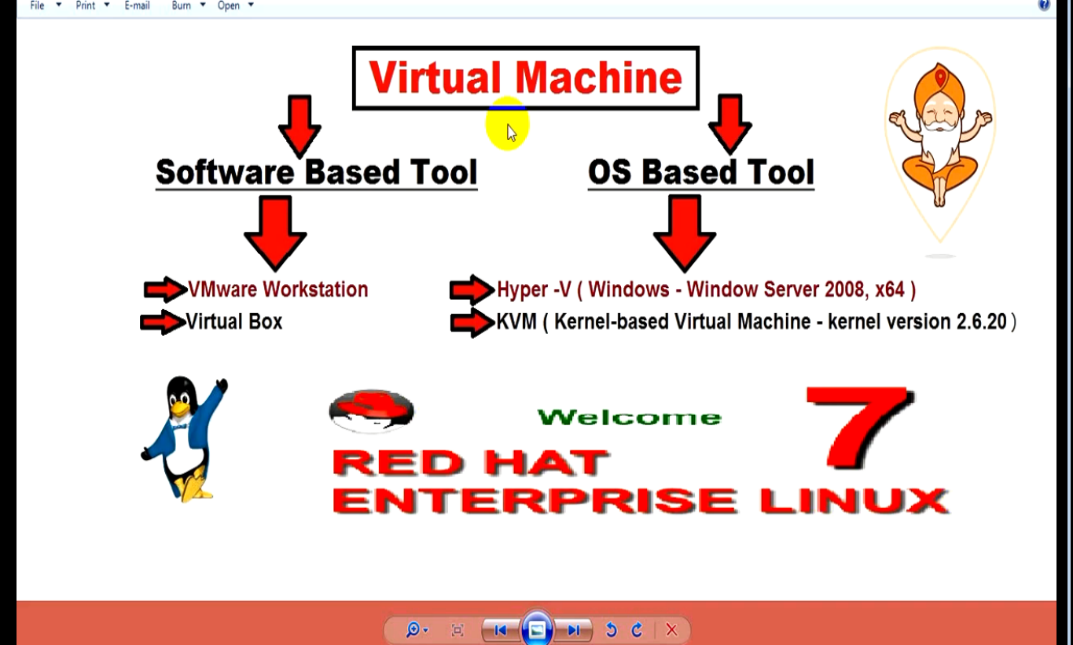
**baseurl=file:///iso (packages directory path)**

**enable=1**

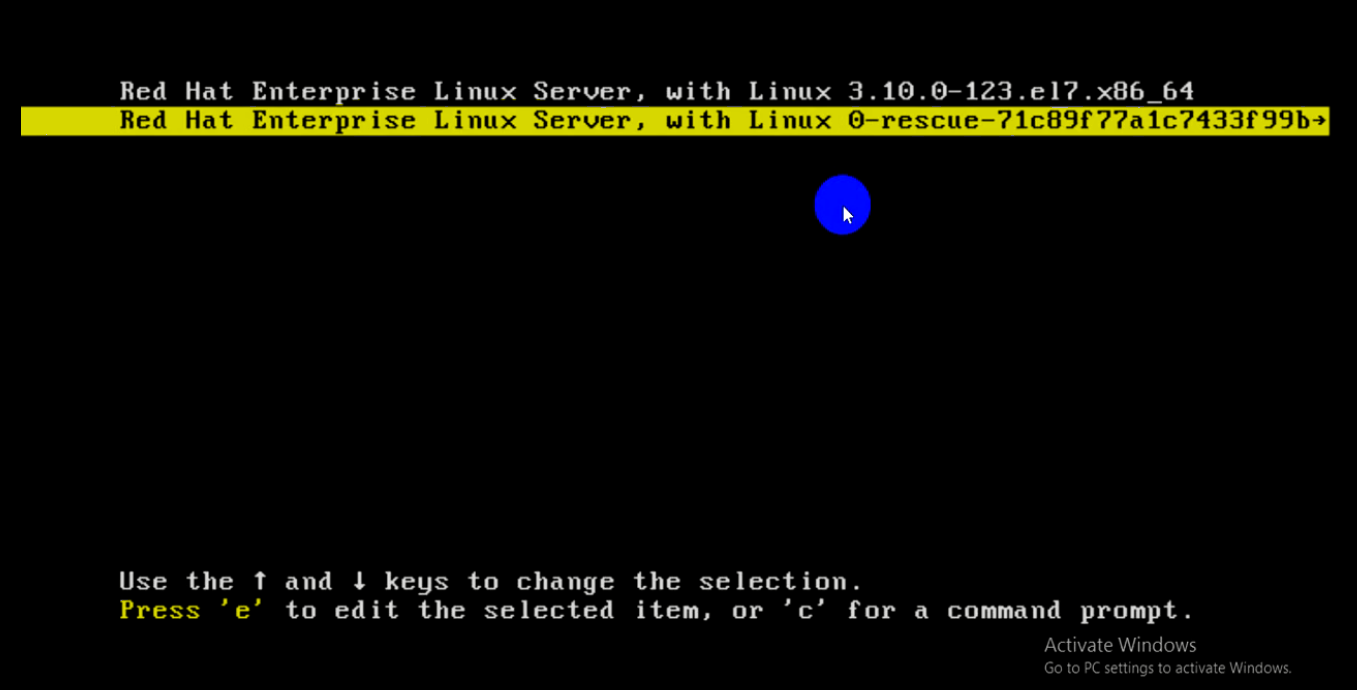
**gpgcheck=0**

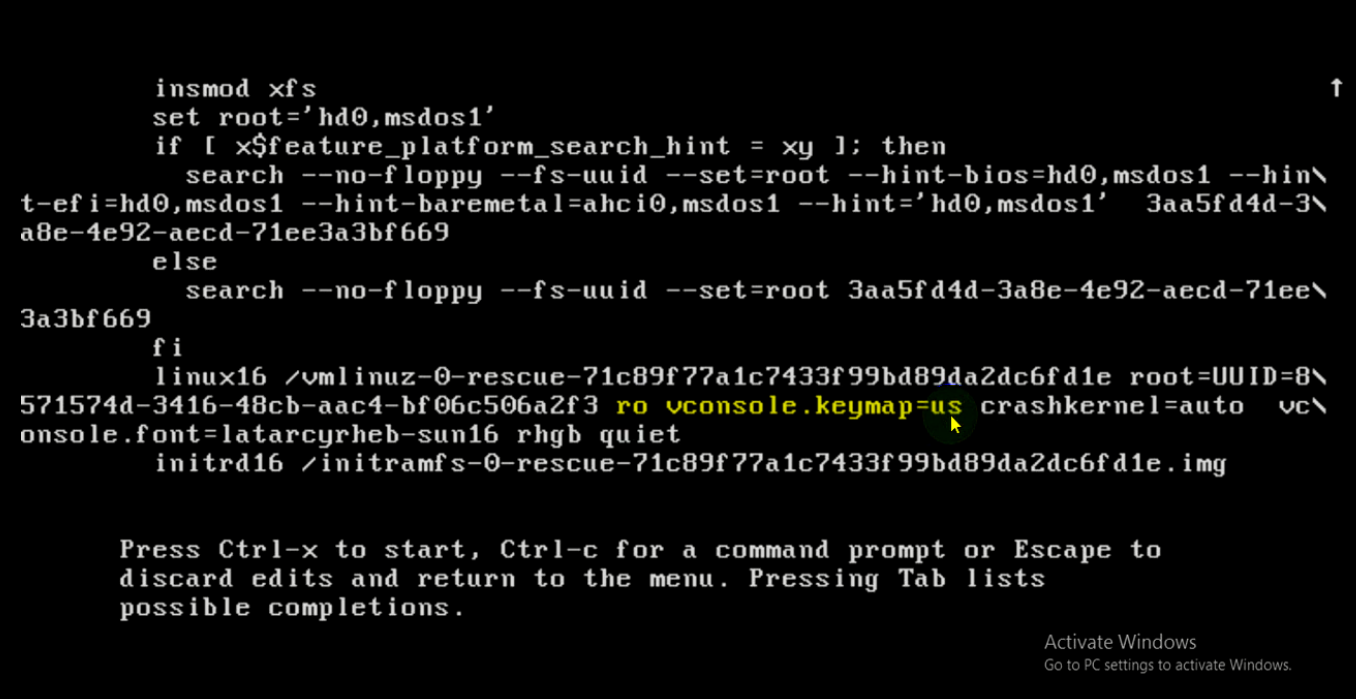
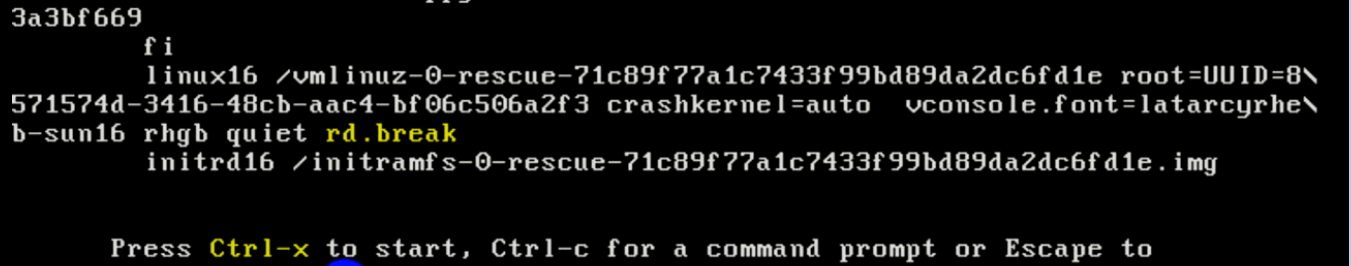
**:wq**

**#yum clear all (clear yum default file )**

**#yum install -y httpd\***

**CRACK Root User Password**

**1 Step: Restart the server system and select 2nd “0-rescue”option and click “e” to start the process.**

**2 Step: Go to down and remove “ro vconsole.keymap=us” then go the last of the line and add “rd.break” then press “Ctrl+x” key.**

**br.break we are using before isoFile loaded we are making brake and assign new password in sysroot**

**3 Step: then automatically one new terminal console will open then type some command.**

**# mount -o remount rw /sysroot (remount and over write sysroot)**

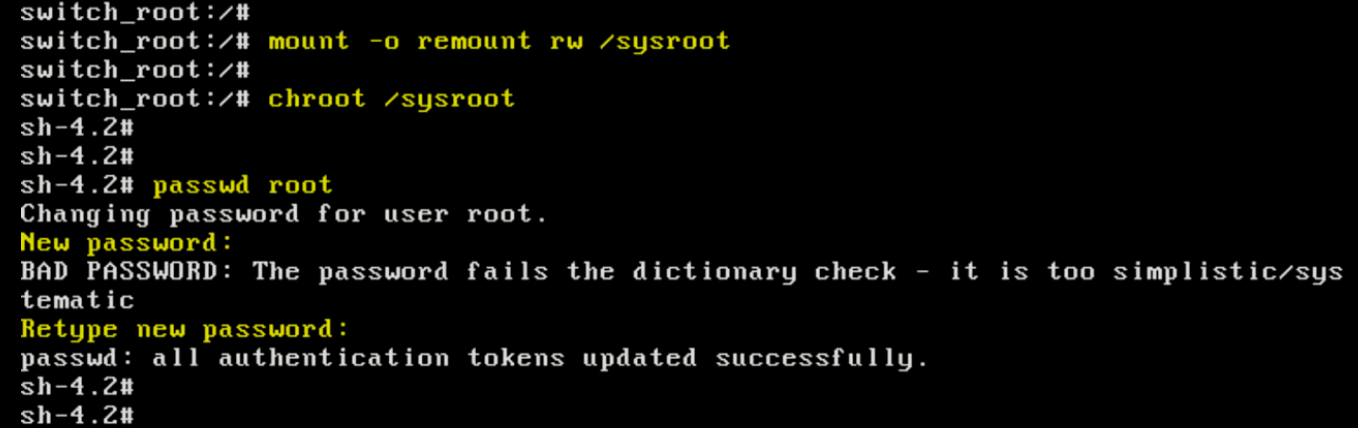
**# chroot /sysroot (change sysroot path)**

**#passwd root**

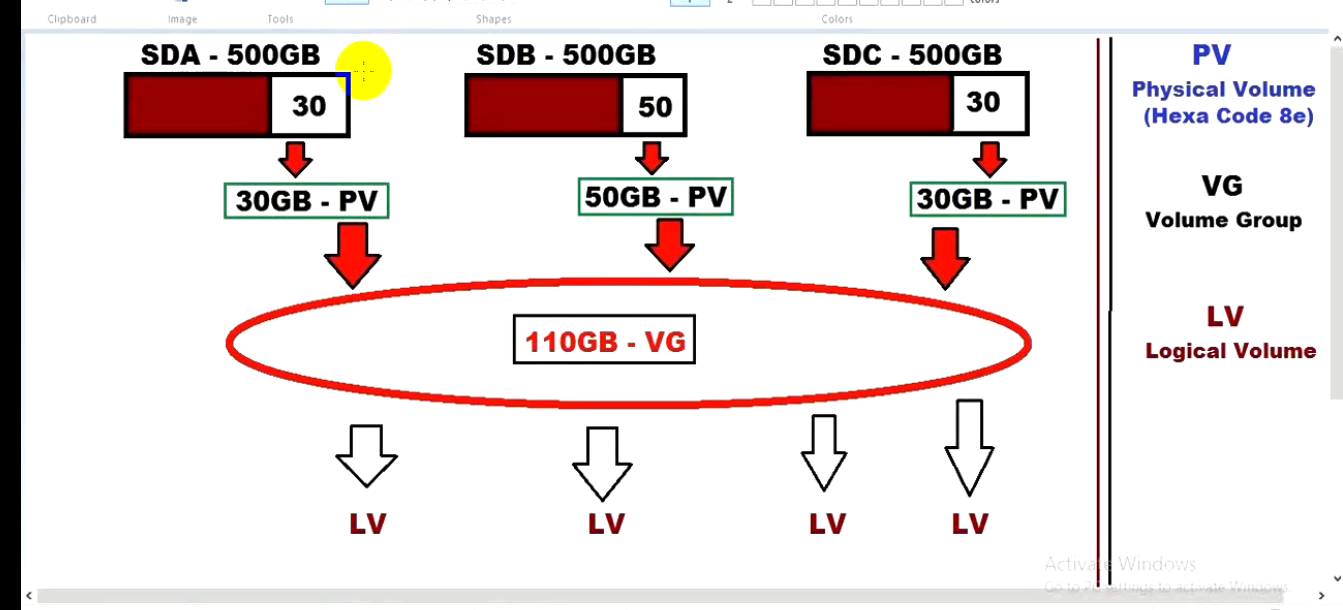
**#touch /.autorelabel**

**#exit**

**#exit**

**After that server system will automatically going to restart and password will changed.**

**LVM partition**



**PV= We can’t extend and format and we can’t mount in any directory.**

**#fdisk /dev/sda (create PV for hard disk and assign 8e hexa code )**

**Cmmd: p (partition create)**

**Select: p (primary partition create)**

**First sector: enter nothing to do**

**Size: +30G**

**Cmmd: p (conformation)**

**Cmmd: t (change hexa code 83 to 8e)**

**Partition number: 4**

**Hexa Code: 8e**

**Cmmd: p (check hexa code chang)**

**Cmmd: w (save)**

**#partprob /dev/sda (configuration in update kernel)**

**#fdisk -l (check)**

**#pvs (PV check)**

**#pvcreate /dev/sda4 (create PV)**

**VG= We can extend the size but we can’t format and mount in any directory.**

**#vgs (check VG)**

**#vgcreate myvg(VGname) /dev/sda4 (create VG)**

**LV= We can format extend and mount in any directory.**

**#lvs (check LV)**

**#lvcreate -L +10G -n mylv(LVname) /sda/myvg(VGpath) (create LV)**

**LVM-Reduce**

**#lvs (display LVM)**

## **#df -Th (Display partition)**

## **#unmount lvname (Unmount the partition)**

## **#vim /etc/fstab (inside configuration file make comment)**

## **#mount -a (Update Mount service)**

## **#e2fsck -f /dev/vgname/lvname (e2fsck Resolved file system error)**

## **#resize2fs -p /dev/vgname/lvname 10G (resize the file system)**

## **#lvreduce -L -10G /dev/vgname/lvname (Reduce the file system)**

## **:yes**

## **##vim /etc/fstab (inside configuration file make Uncomment)**

## **#mount -a (Update Mount service)**

## **#lvs (display LVM)**

## **LVM-Extend**

## **#lvs (display LVM)**

## **#df -Th (Display partition)**

## **#unmount lvname (Unmount the partition)**

## **#vim /etc/fstab (inside configuration file make comment)**

## **#mount -a (Update Mount service)**

## **#lvextend -L +10G /dev/vgname/lvname (Extend the file system)**

## **#e2fsck -f /dev/vgname/lvname (e2fsck Resolved file system error)**

## **#resize2fs /dev/vgname/lvname (resize the file system)**

## **##vim /etc/fstab (inside configuration file make Uncomment)**

## **#mount -a (Update Mount service)**

## **#lvs (display LVM)**

**SSH**(**Secure SHELL**)

**SSH is an open source and most trusted network protocol that is used to login into remote servers for execution of commands and programs. It is also used to transfer files from one computer to another computer over the network using secure copy (SCP) Protocol.**

**#ssh root(username)@192.168.1.100(user machine IP address)**

**: yes(conformation)**

**: wasim123(userpassword)**

**LoginSuccessfullytThroughSSH**

**#vim /etc/ssh/sshd\_config (SSH configuration file)**

**17> Port 22 or 89**

**By default SSH port is 22,we can change using port forwarding but we have to make sure that changing port not using in out network.**

**18> AddressFamily any or 192.168.1.1**

**Any network can access this server but we want only this network users device only can access this server.**

**19> ListenAddress 192.168.1.100**

**We want only one particular device IP address can access this server.**

**47> LoginGraceTime 10m**

**Login time we can set.**

**48> PermitRootLogin no or yes**

**By default ROOT user can login this server through SSH but we can change given NO.**

**49> StrictModes yes or no**

**When we login through SSH that time they asking conformation,that we can change.**

**50> MaxAuthTries 5**

**Password attempt time we can set.**

**51> MaxSession 10**

**We can set at the same time 10 user can login this server through SSH.**

**78> PasswordAuthentication yes or no**

**We can set for any user can login through SSH on this server machine without password authentication.**

**Step 1: Create Authentication SSH-Kegen Keys on – (192.168.1.100)Server Machine**

**Login in with user ROOT we must generate SSH key pass so authentication happens by using this key pass instant of asking user password.**

**SSH keygen is a tool default available in Linux, use to create authentication. ssh-keygen command can create RSA/DSA/TSA/ECDA/ED25519 keys this is are different key encrypted mechanisms.**

**By default SSH key will generate an RSA key pass with one private and public key.**

**RSA default size -b 2048 but it will quickly dump**

First login into server **192.168.1.100** with user **Server Machine** and generate a pair of public keys using following command.

**#ssh-keygen -t rsa -b 4096**

**[Server Machine@192.168.1.100 ~]$ ssh-keygen -t rsa -b 4096**

Generating public/private rsa key pair.

Enter file in which to save the key (/home/tecmint/.ssh/id\_rsa): [Press enter key]

Created directory '/home/tecmint/.ssh'.

Enter passphrase (empty for no passphrase): [Press enter key]

Enter same passphrase again: [Press enter key]

Your identification has been saved in /home/tecmint/.ssh/id\_rsa.

Your public key has been saved in /home/tecmint/.ssh/id\_rsa.pub.

The key fingerprint is:

5f:ad:40:00:8a:d1:9b:99:b3:b0:f8:08:99:c3:ed:d3 tecmint@tecmint.com

The key's randomart image is: IMAGE BELOW

**Step 2: Create .ssh Directory on – (192.168.1.200)Client Machine**

**Use SSH from server 192.168.1.100 to connect server 192.168.1.200 using sshuser as user and create .ssh directory under it, using following command.**

**#ssh sshuser@192.168.1.200 mkdir -p .ssh**

**#ssh sshuser@192.168.1.100 mkdir -p .ssh**

**[Server Machine@192.168.1.100~]$ ssh sshuser@192.168.1.200 mkdir -p .ssh**

The authenticity of host '192.168.0.11 (192.168.0.11)' can't be established.

RSA key fingerprint is 45:0e:28:11:d6:81:62:16:04:3f:db:38:02:la:22:4e.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '192.168.0.11' (ECDSA) to the list of known hosts.

**sheena**@192.168.0.11's password: **sshuser**

**Step 3: Upload Generated Public Keys to – (192.168.1.200)Client Machine**

**#cd /root/.ssh/**

**#ls -l**

**Private key will save in /root/.ssh/id\_rsa**

**Public key will save in /root/.ssh/id\_rsa.pub**

**Use SSH from server 192.168.1.100 and upload new generated public key (id\_rsa.pub) on server 192.168.1.200 under  sshuser .ssh directory as a file name authorized\_keys.**

**#cat .ssh/id\_rsa.pub | ssh sshuser@192.168.1.200 'cat >> .ssh/authorized\_keys'**

**#cat /root/.ssh/id\_rsa.pub | ssh sshuser@192.168.1.100 'cat >> .ssh/authorized\_keys'**

**[Server Machine@192.168.1.100~]$ cat .ssh/id\_rsa.pub | ssh sshuser@192.168.0.11 'cat >> .ssh/authorized\_keys'**

**sshuser**@192.168.1.2's password: **sshuser**

**Step 4: Set Permissions on – (192.168.1.200)Client Machine**

**Due to different SSH versions on servers, we need to set permissions on .ssh directory and authorized\_keys file.**

**#ssh sshuser@192.168.1.200 "chmod 700 .ssh; chmod 640 .ssh/authorized\_keys"**

**#ssh sshuser@192.168.1.100 "chmod 700 .ssh; chmod 640 .ssh/authorized\_keys"**

**[Server Machine@192.168.1.100~]$ ssh sshuser@192.168.0.11 "chmod 700 .ssh; chmod 640 .ssh/authorized\_keys"**

**sshuser**@192.168.0.11's password: **sshuser**

**Step 5: Login from (192.168.1.100)Server Machine to 192.168.1.200)Client Machine Server without Password**

**#ssh sshuser@192.168.1.200**

**#ssh sshuser@192.168.1.100**

**[Server Machine@192.168.1.100~]$ ssh** [sshuser@192.168.1.200](mailto:sshuser@192.168.1.200)

**sshuser**@192.168.0.11's password: **sshuser**

**#exit**

**And try again** user without password.

**MariaDB Configuretion**

**#yum install mariadb\* -y (Install Maria-DB in Server)**

**#systemctl start mariadb (Service Start)**

**#systemctl enable mariadb (Service Enable)**

**#firewall-cmd --permanent --add-service = mysql (Add service in Firewall)**

**#firewall-cmd - -reload (Firewall Reload)**

**#mysql\_secure\_installation (Secure DB Install)**

**Set Password for DB root user: Mysql@123 (Set Password)**

**#mysql -u root -p (Login as a root user)**

**PassSword : Mysql@123 (Password)**

**MDB> show databases; (DB Check)**

**MDB> create databases Wasim; (DB Create)**

**MDB> use Wasim; (Go to DB)**

**MDB> show tables; (DB table show)**

**MDB> describe employees; (DB table open)**

**MDB> drop tables; (DB tables delete)**

**MDB> exit; (bye)**

**MDB> insert into employees (id,name,age,address,salary) values (1,’rahul’,43,’mumbai’,10000.00); (Insert data in DB table)**

**MDB> select \* from employees; (Show emp table data)**

**MDB> select \* from employees where name = ‘rahul’; (Search condition)**

**MDB> select \* from employees**

**where age>10 salary>10000; (Search condition)**

**where age>=10 and salary>10000; (Search condition in and )**

**where age>20 or address = ‘mumbai’; (Search condition in Or)**

**MDB> update employees set address = ‘delhi’ where**

**empid = 2; (Update in DB table chang )**

**MDB> show grants for wasim; (Permission check)**

**MDB> create user eddy@localhost**

**Identified by ‘redhat’ (Create user)**

**MDB> grants create on wasim.\* to eddy@localhost; (Give Permission)**

## **Network**

## **# ifconfig (IP check)**

## **# ifconfig enp0s3 (IP check with device name)**

## **# nmcli (Configure IP)**

## **#nmcli connection add con-name wasim ifname enp0s3 type ethernet autoconnect yes ip4 ‘192.168.1.100/24’ (Configure IP address)**

## **# nmcli con show (IP Connection check)**

## **# nmcli connection modify wasim ip4 192.168.1.100/24 (Modify IP address)**

## **# ifup enp0s3 (LAN connection UP)**

## **# ifdown enp0s3 (LAN connection DOWN)**

## **# vim /etc/sysconfig/network-scripts/ifcfg-enp0s3 (Network configuration file)**

## **3 BOOTPORT=static**

## **16 ONBOOT=yes**

## **17 IP ADDR=192.168.1.100**

## **18 NETMASK=255.255.255.0**

## **: wq(save)**

## **# ifup enp0s3**

## **# ifdown enp0s3**

**# init (Linux 1st proses)**

**# ps -ef (Running proses check)**

**# netstat -tulpn (Linux Ports check)**

**# top (Memory utilization check)**

**# free -m (RAM Memory check)**

**# who -r (Current runlevel check)**

**# runlevel**

**# fdisk -l (Partition table check)**

**# df -Th (Partition check)**

**# /etc (System configuration file save)**

**# cat /proc/lodeavg (System Loadavarage check)**

**# w**

**# uptime**

**# uname -a (Kernel version check)**

**# uname -r**

**Backup user ID**

## **# mkdir /backup (create a dir for user backup)**

## **# tar -cvfz /backup/user.bak.tar.gz/home/”username”**

## **C=Create a new .tar archive file**

## **V=Verbosely show the .tar file**

## **F=File name type**

## **Z=Uncompress file**

## **File System In LINUX**

## **Ext2 Ext3 Ext3 Ext4 JFS XFS Btrfs FAT etc.**

## **SELinux**

## **Security-enhanced linux is a linux kernel security module that provides a mechanism for supporting access control security policies.**

## **# cd /etc**

## **Etc]# ls -lZr passwd (SELinux Context)**

## **-rw-rr-r--.1 root root system\_u: object\_r: password\_file\_t:s0**

## **System\_u=SELinux user**

## **Object\_r=SELinux role**

## **Passwd\_file\_t=SELinux type**

## **S0=Sensitivity level**

## **There is 3user in linux A1,B1 and C1,give access for A1 user FTP,TELNET and SSH,for B1 user SSH,for C1 user TELNET.**

## **Configure sudoers file for privileged**

## **# vim /etc/sudoers**

## **Add below line-------**

## **A1 ALL=NOPASSWD: /bin/ftp, /bin/telnet, /bin/ssh**

## **B1 ALL=NOPASSWD: /bin/ssh**

## **C1 ALL=NOPASSWD: /bin/telnet**

## **:wq(save)**

## **FireWall**

## Firewall is a nerwork security system that filters and controls the traffic on predetermined set of rules.

## Firewall in Linux work besed on net filters kernel module.

## **FSTAB**

## # cat /etc/fstab

## 1./dev/sad2 2./boss 3.xfs 4.default 5.00

* **Device** – the first field specifies the mount device. These are usually device filenames. Most distributions now specify partitions by their labels or UUIDs.
* **Mount point** – the second field specifies the mount point, the directory where the partition or disk will be mounted. This should usually be an empty directory in another file system.
* **File system type** – the third field specifies the file system type.
* **Options** – the fourth field specifies the mount options. Most file systems support several mount options, which modify how the kernel treats the file system. You may specify multiple mount options, separated by commas.
* **Backup operation** – the fifth field contains a **1** if the**dump** utility should back up a partition or a **0** if it shouldn’t. If you never use the dump backup program, you can ignore this option.
* **File system check order** – the sixth field specifies the order in which *fsck* checks the device/partition for errors at boot time. A **0** means that fsck should not check a file system. Higher numbers represent the check order. The root partition should have a value of **1** , and all others that need to be checked should have a value of **2**.

## **GRAP || PIPE || SORT || XML**

## GRAP# grep “jone” test.txt

## # grep -w “jone” test.txt (By word search)

## # grep -wi “jone” test.txt (By K-sensitive search)

## # grep -win “jone” test.txt (By with line number search)

## # grep -B 4 “jone” test.txt (By befor 4 line show)

## # grep -A 4 “jone” test.txt (By after 4 line show)

## # grep -C 3 “jone” test.txt (befor and after 3 line show)

## # grep -win “jone” ./\* (in current directory search)

## # grep -win “jone” ./\*.txt (in current directory .txt file search only)

## # grep -win “jone” ./ (in current directory and sub dir all search)

## # grep -win “jone” . (Only path)

## # grep e/f/c test.txt (by word search only )

## PIPE #cat test.txt | grep -v a | short -r (using pipe to command run)

## SORT# sort test.txt (line by line)

## # sort test.txt > sample.txt (copy all contain)

## XML# less test.xml (to check line by line)

## # sed “ “ test.xml (All data show)

## # sed “//,//” test.xml (using string)

## <group>

## <name>IBM</name>

## <uid>1</uid>

## </group>

## # sed “/group/,/\/group/d” test.xml (by <group> to </group> show)

## # sed “/group/,/\/group/d” test.xml > user.xml (sent only view data)

## # sed “/group/,/\/group/s/name/groupname/g” test.xml (content change in side the data g for globally)

## # sed -i “/group/,/\/group/s/name/groupname/g” test.xml (permanent change)

## # xmlint --schema shema.xsd test.txt (verify XML file)

## **CRON Tab**

## # vim /etc/corntab (Cron tab configuration file)

## 1 \* MM - Only minuets

## 2 \* HOURS - Hour per day

## 3 \* DOM - date of month

## 4 \* MON - Month of name

## 5 \* DAY – Day of week

## CMD – Command and script.

## **MEMORY || CPU || H/W**

## # cat /proc/meminfo | grep MemTotal (H/W check)

## # cat /proc/cpuinfo (CPU check)

## # dmidecode -s system-serial-number (S/L No.Model No.check)

## # sar (CPU utilization history check)

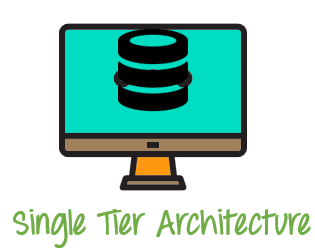
## **If SGID bit is set on any directory, all subdirectories and files created inside will get same group ownership as the main directory, it doesn’t matter who is creating.**

## # chmod 2775 /usr/sbin/useradd (Remove SGID)

**What is Database Architecture?**

Database architecture uses programming languages to design a particular type of software for businesses or organizations.. A database stores critical information for a business. Selecting the correct Database Architecture helps in quick and secure access to this data.

**1 tier Architecture**

[](https://www.guru99.com/images/1/091318_0745_DBMSArchite1.png)

1-tier Architecture Diagram

The simplest of Database Architecture are **1 tier**where the Client, Server, and Database all reside on the same machine. Anytime you install a DB in your system and access it to practise SQL queries it is 1 tier architecture. But such architecture is rarely used in production.

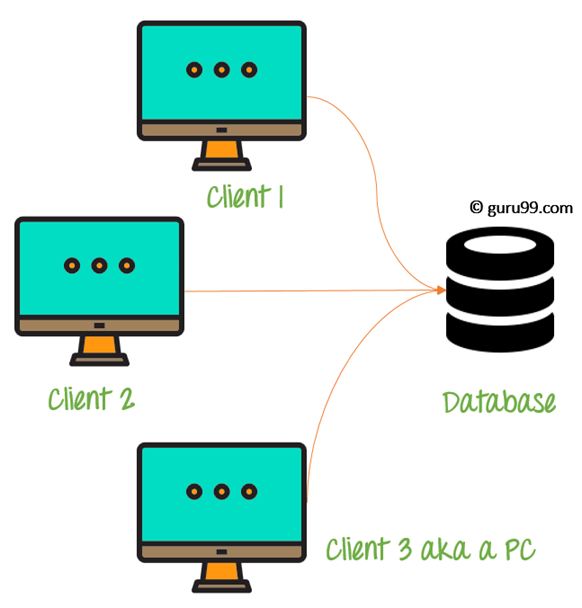
**2-tier Architecture**

A two-tier architecture is a database architecture where

1. Presentation layer runs on a client (PC, Mobile, Tablet, etc)
2. Data is stored on a Server.

An application interface which is called ODBC (Open Database Connectivity) an API which allows the client-side program to call the DBMS. Today most of the DBMS offers ODBC drivers for their DBMS. 2 tier architecture provides added security to the DBMS as it is not exposed to the end user directly.

**Example** of Two-tier Architecture is a Contact Management System created using MS- Access.

[](https://www.guru99.com/images/1/091318_0745_DBMSArchite2.png)

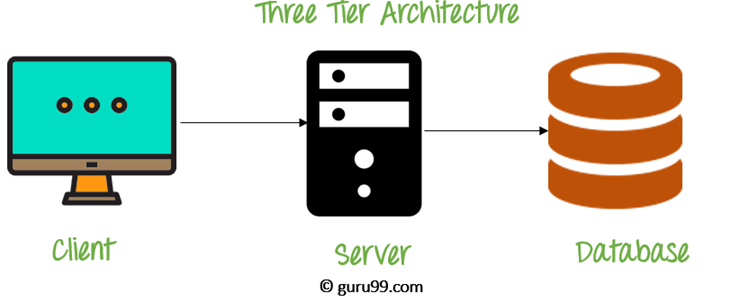
2-tier Architecture Diagram

In the above 2-teir architecture we can see that one server is connected with clients 1, 2m and 3. This architecture provides Direct and faster communication.

**3-tier Architecture**

3-tier schema is an extension of the 2-tier architecture. 3-tier architecture has following layers

1. Presentation layer (your PC, Tablet, Mobile, etc.)
2. Application layer (server)
3. Database Server

[](https://www.guru99.com/images/1/091318_0745_DBMSArchite3.png)

3-tier Architecture Diagram

This DBMS architecture contains an Application layer between the user and the DBMS, which is responsible for communicating the user's request to the DBMS system and send the response from the DBMS to the user.The application layer(business logic layer) also processes functional logic, constraint, and rules before passing data to the user or down to the DBMS Three tier architecture is the most popular DBMS architecture.

**The goal of Three-teir architecture is:**

* To separate the user applications and physical database
* Proposed to support DBMS characteristics
* Program-data independence
* Support of multiple views of the data

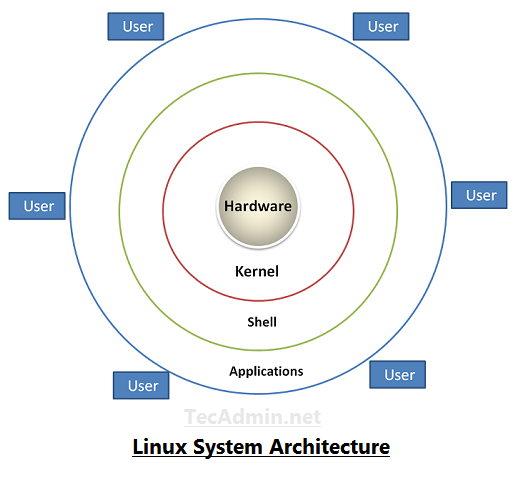
Example of Three-teir Architecture Any large website on the internet, including guru99.com ??

**Summary**

* DBMS architecture helps in design, development, implementation, and maintenance of a database
* The simplest of Database Architecture are 1 tier where the Client, Server, and Database all reside on the same machine
* A two-tier architecture is a database architecture where presentation layer runs on a client and .data is stored on a Server
* 3-tier architecture is consist of the Presentation layer (PC, Tablet, Mobile, etc.), the Application layer (server) and Database Server

# **Linux Architecture**

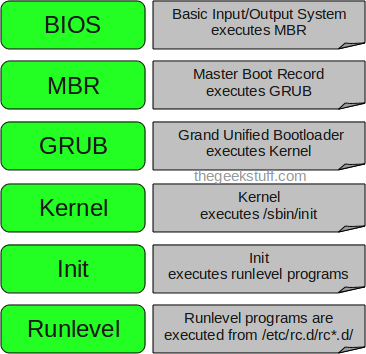
The Linux system works basically on 4 layers. First view the diagram below, which is showing the architecture of a Linux System.

[](https://tecadmin.net/tutorial/wp-content/uploads/2017/10/linux-architecture-image.png)

* **Hardware −** Hardware consists of all physical devices attached to the System. For example: Hard disk drive, RAM, Motherboard, CPU etc.
* **Kernel −** Kernel is the core component for any (Linux) operating system which directly interacts with the hardware.
* **Shell −** Shell is the interface which takes input from Users and sends instructions to the Kernel, Also takes the output from Kernel and send the result back to output shell.
* **Applications −** These are the utility programms which runs on Shell. This can be any application like Your web browser, media player, text editor etc.

# **Linux Boot Process**

Press the power button on your system, and after few moments you see the Linux login prompt.

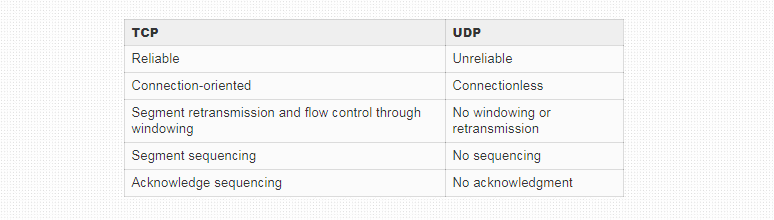
The following are the 6 high level stages of a typical Linux boot process.  
  


### **1. BIOS**

* BIOS stands for Basic Input/Output System
* Performs some system integrity checks
* Searches, loads, and executes the boot loader program.
* 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.
* 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
* 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
* 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

### **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/



**20.How to reduce load average of CPU shown in top command?**

check out for no. of sleeping and zombie process in your server for which I would advice to kill all not required running processes

Code:

Tasks: 641 total, 2 running, 628 sleeping, 0 stopped, 11 zombie

There are few factors that determine the Load Average of the Server Load. If the server uses dual processors the acceptable server load average is 2.00. This load is considered optimal.   
Give the command "top" and check which process is utilizing more resources.   
Check for mainly these processes.   
  
1)sendmail 2)mysql 3)java.   
  
These applications running status depend upon the applications you install and use them.   
Most of the times sendmail process will run multiple times. So once restart the sendmail process to make it normal.  
  
/etc/init.d/sendmail restart

**Understand Linux Load Averages and Monitor Performance of Linux**

1. System **load**/CPU **Load** – is a measurement of CPU over or under-utilization in a Linux system; the number of processes which are being executed by the CPU or in waiting state.
2. **Load average** – is the **average** system **load** calculated over a given period of time of 1, 5 and 15 minutes.

**4 different commands to check the load average in linux**

1. Command 1: Run the command, “cat /proc/loadavg” .
2. Command 2 : Run the command, “w” .
3. Command 3 : Run the command, “uptime” .
4. Command 4: Run the command, “top” . See the first line of top command's output.

**17.If in load average all 3 fields shown as 4 4 4 what does it mean?**

There are numerous ways of monitoring system load average including **$ uptime** which shows how long the system has been running, number of users together with load averages:

**$ uptime**

07:13:53 up 8 days, 19 min, 1 user, load average: 1.98, 2.15, 2.21

The numbers are read from left to right, and the output above means that:

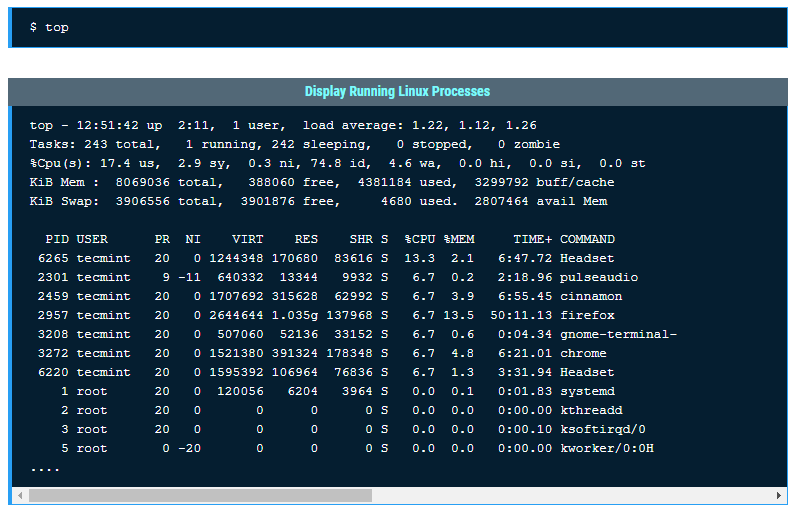
* load average over the last 1 minute is 1.98
* load average over the last 5 minutes is 2.15
* load average over the last 15 minutes is 2.21

High load averages imply that a system is overloaded; many processes are waiting for CPU time.

**The three fields show the load average over the last 1, 5 and 15 minutes respectively. Therefore, if the load average for all fields is 4, it means it has been constantly at 4% load for at 15 minutes or more.**

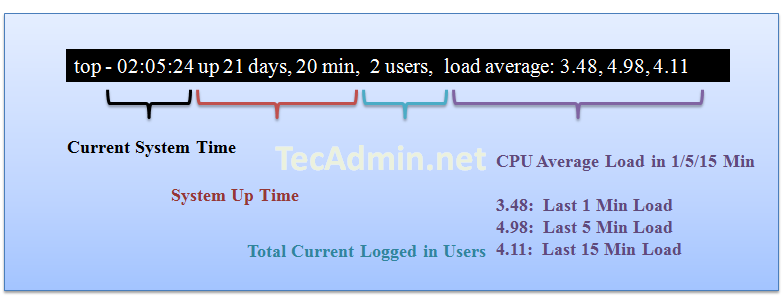
**16.In top command what does 3 numeric field indicates?**

**When you execute the $ top command on Linux, it shows a lot of results, here I am trying to show you to how to read it row by row.**



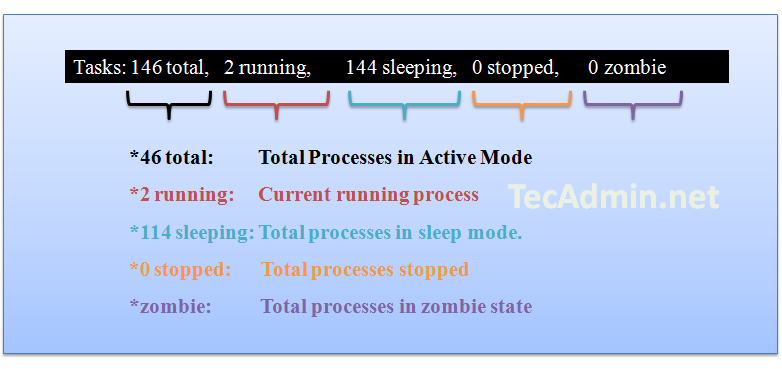
**Result Row #1**

**Row 1 results show about server uptime from the last reboot, currently logged in users and CPU load on the server. The same output you can find using Linux uptime command.**



**Result Row #2**

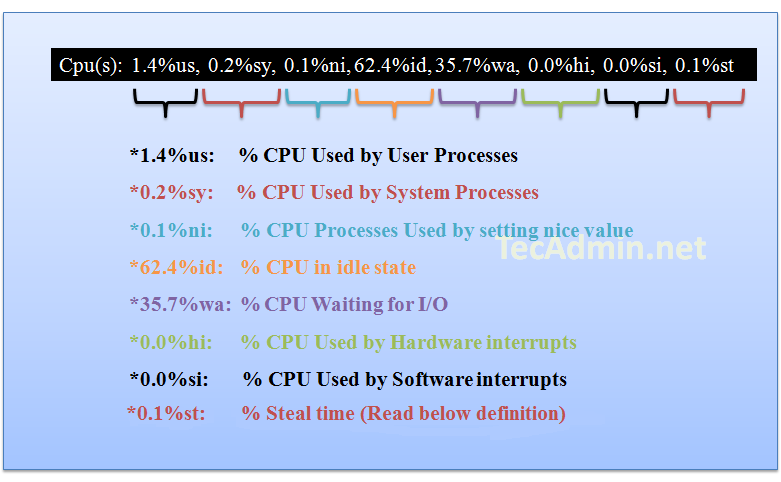
**Row 2 shows the number of process running on the server and their state.**



**Zombie process or defunct process is a process that has completed execution but still has an entry in the process table. This entry is still needed to allow the parent process to read its child’s exit status.**

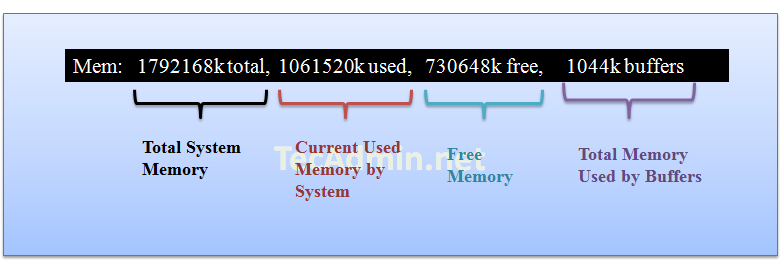
**Result Row #3**

**Row three shows the CPU utilization status on the server, you can find here how much CPU is free and how much is utilizing by the system.**



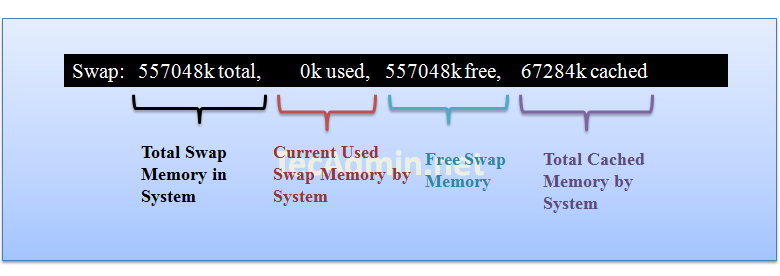
**Result Row #4**

**Row 4 shows the memory utilization on the server, you can find here how much memory is used, the same results you can find using free command.**



**Result Row #5**

**Row 5 shows the**[**swap memory**](https://tecadmin.net/enable-swap-on-ubuntu/)**utilization on the server, you can find here how much swap is being used, the same results you can find using free command.**



**Result Row #6 ( Running Processes )**

**In this steps you will see all running process on servers and there additional details about them like below.**