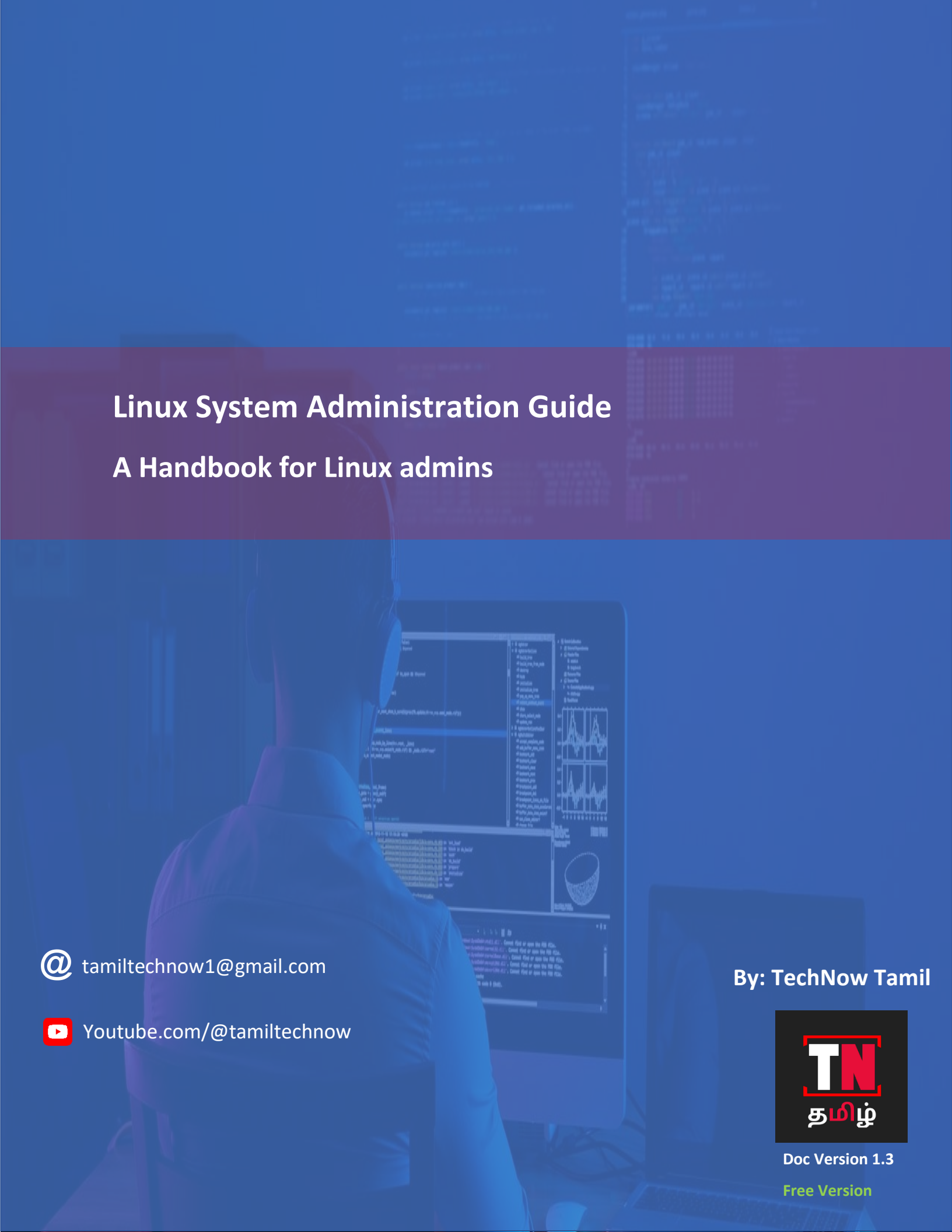
Linux System Administration Guide

A Handbook for Linux admins



@ tamiltechnow1@gmail.com

Youtube.com/@tamiltechnow

By: TechNow Tamil

Doc Version 1.3

Free Version

Linux System Administration Guide

By TechNow Tamil

Abstract:

This book is meant to be used for self-study; the intent is to read this book next to a working Linux computer so you can immediately do every subject & practice each command.

This book is aimed at beginner Linux system administrators and might be interesting and useful for intermediate users that want to know a bit more about their Linux system. However, if you think of other ideas that can enrich this e-book, feel free to drop us a note at one of our social network profiles:

Youtube : www.youtube.com/@tamiltechnow

Gmail : tamiltechnow1@gmail.com

Telegram : https://t.me/deepak_d1

Click Here to Subscribe TechNow Tamil : Subscribe

Author: Deepak

Important Note:

In this E-book the word **RedHat Based distros** means, those commands will work on following Operating systems, RHEL, Oracle Linux, CentOS, Alma Linux, Fedora, Rocky Linux.

In this E-book the word **Debian Based distros** means, those commands will work on following Operating systems, Ubuntu, Linux Mint, Kali Linux, Debian.

Note: Contents of this will be modified/Updated frequently for enhancing this book.

Table of Contents:

Chapter 1: Basic Linux Commands

Linux provides a CLI (Command Line Interface) to communicate with the OS. Here are the most basic Linux Commands.

1. pwd

Print Working Directory, PWD Display the path of the current working directory inside the terminal.

Syntax:

```
pwd
```

```
root@tn:~# pwd
/root
```

2. touch

The touch command in Linux is used to create new files without any content inside it (Empty Files).

Syntax:

```
touch <File Name>
```

Command	Explanation
touch file1 file2	Creates 2 files (file1 & file2) at a time in current directory.
touch file{1..5}	Creates 5 files (file1 to file5) at a time in current directory.

Examples:

The following example creates **file1.txt** in current working directory.

```
touch file1.txt
```

```
root@tn:~# touch file1.txt
root@tn:~# ls
file1.txt
```

The following example, creates **file1.txt** and **file2.txt** in present directory.

```
touch file1.txt file2.txt
```

```
root@tn:~# touch file1.txt file2.txt
root@tn:~# ls
file1.txt  file2.txt
```

The following example creates **file1.txt, file2.txt, file3.txt, file4.txt** and **file5.txt** in present directory.

```
touch file{1..5}.txt
```

```
root@tn:~# touch {1..5}.txt
root@tn:~# ls
1.txt  2.txt  3.txt  4.txt  5.txt
```

3. cat

The cat command in Linux is used to read the contents of one or more files and display

their contents inside the terminal.

Syntax:

```
cat <file name>
```

Command	Explanation
cat -b	This is used to add line numbers to non-blank lines

Examples:

The following example shows the contents of file1.txt file.

```
cat file1.txt
```

```
root@tn:~# cat file1.txt
Hi Mate,
This is file1,
Thanks.
```

The following example shows contents of **file1.txt** file with line number.

```
cat -b file1.txt
```

```
root@tn:~# cat -b file1.txt
     1  Hi Mate,
     2  This is file1,
     3  Thanks.
```

4. mkdir

This command used to Create directories (folders) in linux / unix operating systems.

Syntax:

```
mkdir (directory name)
```

Command	Explanation
mkdir -p	Creates both a new parent directory and a sub-directory

Examples:

The following example creates **mydocs** directory in current working directory.

```
mkdir mydocs
```

```
root@tn:~# mkdir mydocs
mkdir: created directory 'mydocs'
```

Consider we have to create **document** directory first and inside document directory need to create **tech** directory.

We can create both parent & sub directory in single command with **-p** option.

The following command, creates **document** directory first & inside **document** directory creates **tech** directory.

```
mkdir -p document/tech
```

```
root@tn:~# mkdir -p document/tech
root@tn:~# ls document/
tech
```

5. cd

The cd command in linux expands to 'Change Directory'. We can move/navigate between directories in linux / unix operating system.

Syntax:

```
cd (directory name)
```

Command	Explanation
cd ~	This command used to go to home directory of current user from any path
cd /	Changes the directory to / directory
cd ..	Change Working Directory to previous folder (Ex: [/var/log] to [/var])

Examples:

The following example, Change working directory to **document** directory

```
cd document
```

```
root@tn:~# cd document/
root@tn:~/document# pwd
/root/document
```

The following command changed the working directory from **/document** to **/root** (root user home directory is /root).

Using **~** symbol we can change working directory to current user's home directory.

```
cd ~
```

```
root@tn:~/document# cd ~
root@tn:~# pwd
/root
```

The following example using **/** symbol we are changing working directory to **/**.

```
cd /
```

```
root@tn:~# cd /
root@tn:/# pwd
/
```

To come back to previous directory type **..** symbol. The following example we back from **/root/document** to **/root** directory.

```
cd ..
```

```
root@tn:~/document# cd ..
root@tn:~# pwd
/root
```

6. ls

The ls command used to list all contents available in a directory.

Syntax:

```
ls [Option]
```

Command	Explanation
ls -l	lists all the contents along with its owner settings, permissions & time stamp (long format)
ls -a	lists all the hidden contents in the specified directory
ls -lS	Option -S, sorts and lists all the contents in the specified directory by size
ls -lh	Option -h, list all the contents with size in human readable format
ls -lt	Option -t, sorts and lists all the contents in the specified directory by time
ls *.txt	Using '*' flag, lists only the contents in the directory of a particular format. (Ex: .txt, .pdf)

Examples:

List contents of current working directory simply (without properties of the contents).

```
ls
```

```
root@tn:~# ls
document
```

The following command will list the contents of **document** directory.

```
ls document/
```

```
root@tn:~# ls document/
tech
```

The following command will list the contents of current directory with properties of the contents.

```
ls -l
```

```
root@tn:~# ls -l
total 4
drwxr-xr-x 3 root root 4096 Dec 28 09:49 document
```

The following command will list the contents & hidden contents of current working directory.

```
ls -a
```

```
root@tn:~# ls -a
.   .bash_history  .cache    .gitconfig      .profile  .viminfo
..  .bashrc        document  .mysql_history  .ssh      .wget-hsts
```

The following command will list the contents of current directory with details and sort by file size.

```
ls -lS
```

```
root@tn:~# ls -ls
total 12
4 drwxr-xr-x 3 root root 4096 Dec 28 09:49 document
4 drwxr-xr-x 3 root root 4096 Dec 27 14:56 mydocs
4 -rw-r--r-- 1 root root 1581 Dec 28 11:07 test.txt
```

The following command will list the contents of current directory with details and show the file size in human readable format.

```
ls -lh
```

```
root@tn:~# ls -lh
total 12K
drwxr-xr-x 3 root root 4.0K Dec 28 09:49 document
drwxr-xr-x 3 root root 4.0K Dec 27 14:56 mydocs
-rw-r--r-- 1 root root 1.6K Dec 28 11:07 test.txt
```

The following command will list the *contents of current directory with details and sort by timestamp (newest first).*

```
ls -lt
```

```
root@tn:~# ls -lt
total 12
-rw-r--r-- 1 root root 1581 Dec 28 11:07 test.txt
drwxr-xr-x 3 root root 4096 Dec 28 09:49 document
drwxr-xr-x 3 root root 4096 Dec 27 14:56 mydocs
```

The following command will list, only the files end with **.log** in **/var/log/** directory.

```
ls -l /var/log/*.log
```

```
root@tn:~# ls -l /var/log/*.log
-rw-r--r-- 1 root   root   17160 Dec 27 17:21 /var/log/alternatives.log
-rw-r----- 1 syslog adm    13835 Dec 28 11:17 /var/log/auth.log
-rw-r--r-- 1 root   root   57457 Feb 27  2019 /var/log/bootstrap.log
```

Similarly, you can list any type of files.

Ex: ls -l /var/log/*.txt → will list only the files end with **.txt**

```
ls -l /var/log/*.txt
```

```
root@tn:~# ls -l /var/log/*.txt
```

7. cp

The cp command in Linux translates to 'copy'. It is used to copy files/directories from one location to another from inside the terminal.

Syntax:

```
cp [options] [Arguments:(file name) (destination path)]
```

Command for copy files:

```
cp (file name) (destination path)
```

Command for copy directories: (Directories must be copied with -r option)

```
cp -r (directory name) (destination path)
```

Command	Explanation
cp -r	Recursive copy for copying directories; Copies even hidden files
cp -i	Enters interactive mode; CLI asks before overwriting files
cp -n	Does not overwrite the file
cp -u	Updates the destination file only when the source file is different
cp -v	Verbose; Prints informative messages

Examples:

The following example copy **file1.txt** to **/tmp** directory.

```
cp file1.txt /tmp/
```

```
root@tn:~# cp file1.txt /tmp/
```

The following example will copy **file1.txt** to **/tmp** directory and shows the Output.

```
cp -v file1.txt /tmp/
```

```
root@tn:~# cp -v file1.txt /tmp/
'file1.txt' -> '/tmp/file1.txt'
```

The following command with -r option, will copy **mydoc1** (directory) to **/tmp/** directory.

```
cp -r mydoc1/ /tmp/
```

```
root@tn:~# cp -r mydoc1/ /tmp/
```

If you didn't give **-r** option, you will get following error.

```
root@tn:~# cp mydoc1/ /tmp/
cp: omitting directory 'mydoc1/'
```

The following example, If same file already available in destination CLI asks before overwriting the file.

```
cp -i file1.txt /tmp/
```

```
root@tn:~# cp -i file1.txt /tmp/
cp: overwrite '/tmp/file1.txt'?
```

The following example, If same file already available in destination **-n** does not overwrite the file.

```
cp -n file1.txt /tmp/
```

```
root@tn:~# cp -n file1.txt /tmp/
```

The following example, the command updates the destination file only when the source file is different from the destination file.

```
cp -u file1.txt /tmp/
```

```
root@tn:~# cp -u file1.txt /tmp/
```

8. mv

Moves files and directories from one directory to another.

It performs two major functions in Linux.

- You can easily move a file/directory from one location to another.
- You can rename a file/directory using this command.

Syntax:

```
mv <options> (file name) (destination path)

Syntax for move files and directories
mv (file name (or) directory name) (destination path)

Syntax for rename files & directories
mv (Old file name (or) directory name) (new file name (or) directory name)
```

Command	Explanation
mv -i	Enters interactive mode; CLI asks before overwriting files
mv -v	Verbose; Prints informative messages

Examples:

The following command will move **test.txt** file to **mydocs/** directory.

```
mv test.txt /mydocs/
```

```
root@tn:~# mv test.txt mydocs/
root@tn:~# ls mydocs/
test.txt
```

The following example, **-i** option, asks before overwriting the file If same file already available in destination.

```
mv -i test.txt /folder1/
```

```
root@tn:~# mv -i test.txt mydocs/
mv: overwrite 'mydocs/test.txt'?
```

The following example will move the **dock.txt** file to **mydocs/** directory and prints the output.

```
mv -v dock.txt /folder1/
```

```
root@tn:~# mv -v dock.txt mydocs/
'dock.txt' -> 'mydocs/dock.txt'
```

The following example renames the **test.txt** name to **file.txt**.

```
mv test.txt file.txt
```

```
root@tn:~# mv test.txt file.txt
root@tn:~# ls
file.txt
```

9. rm

The rm command in Linux helps you to delete files and directories.

This is very dangerous command which could cause data loss. So always use carefully.

Syntax:

```
rm [option] (File Name)
```

Command	Explanation
rm -r	Option -r, Used to delete directories with all available contents
rm -rf	Option -f, Used to delete contents without confirmation

Examples:

The following command Deletes the file named **file.txt**.

```
rm file.txt
```

```
root@tn:~# rm file.txt
```

The following example, **rm** command with **-r** can delete the directory.

```
rm -r docs
```

```
root@tn:~# rm -r document
```

Without **-r** option rm command cannot delete directories.

```
root@tn:~# rm document
rm: cannot remove 'document': Is a directory
```

10. rmdir

rmdir command is used remove empty directories from the filesystem in Linux. The rmdir command removes specified directory only if the directory is empty.

So, if the specified directory has some directories or files in it then this cannot be removed by rmdir command.

Syntax:

```
rmdir (Empty Directory Name)
```

Examples:

The following command, Deletes docs directory because docs directory is empty.

```
rmdir docs
```

```
root@tn:~# rmdir docs
root@tn:~#
```

If the directory is not empty you can see below error.

```
root@tn:~# rmdir docs
rmdir: failed to remove 'docs': Directory not empty
```

11. grep

The grep command in Linux searches through a specified file and prints all lines that match a given

Word or pattern. We can find words in any files with this command.

Syntax:

```
grep [option] (File Name)
```

Command	Explanation
grep -i	Returns the results for case insensitive pattern \ word
grep -n	Display the matched lines and their line numbers
grep -c	Returns the count of the lines that match a pattern \ word
grep -v	This prints out all the lines that do not matches the pattern \ word

Examples:

For example, **test.txt** contains following lines.

```
root@tn:~# cat test.txt
This is a text file.
This text file name is test.txt.
This text file is available in /root directory.
Its not available in /var directory.
```

The following example, we are finding lines that contains word **"dir"**.

```
grep dir test.txt
```

```
root@tn:~# grep dir test.txt
This text file is available in /root directory.
Its not available in /var directory.
```

We can find case sensitive words using grep command with **-i** option.

```
grep -i Dir test.txt
```

```
root@tn:~# grep -i Dir test.txt
This text file is available in /root directory.
Its not available in /var directory.
```

Without **-i** option it will not return matching words. Because no word is starting with capital **D** for **directory** word.

```
root@tn:~# grep Dir test.txt
root@tn:~#
```

The following example, we are finding lines that contains word **"dir"** with line number.

```
grep -n dir test.txt
```

```
root@tn:~# grep -n dir test.txt
3:This text file is available in /root directory.
4:Its not available in /var directory.
```

Grep command with **-c** option will print only the count of lines that contains word **"dir"**.

```
grep -c dir test.txt
```

```
root@tn:~# grep -c dir test.txt
2
```

Grep command with **-v** option will print only the lines that's do not contain word **"dir".**

```
grep -v dir test.txt
```

```
root@tn:~# grep -v dir test.txt
This is a text file.
This text file name is test.txt.
```

12. head

The head command in Linux prints the first 10 lines of a given file content. You can specify number of lines want to view / print by -n option.

Syntax:

```
head [option] (File Name)
```

Command	Explanation
head -n	Print the first n lines instead of the first 10;

Examples:

Assume there is a file named file.txt which contains 20 Lines.

The following command print the first 10 lines of the file by default.

```
head file.txt
```

```
root@tn:~# head file.txt
This is line 1.
This is line 2.
This is line 3.
This is line 4.
This is line 5.
This is line 6.
This is line 7.
This is line 8.
This is line 9.
This is line 10.
```

The following command print the first 2 lines of the file.

```
head -2 file.txt
```

```
root@tn:~# head -2 file.txt
This is line 1.
This is line 2.
```

13. tail

The tail command in Linux prints the last **10** lines of a given file content. You can specify number of lines want to view / print by -n option.

Syntax:

```
tail [option] (File Name)
```

Command	Explanation
tail -n	Print the last n lines instead of the last 10
tail -f	This option shows the last ten lines of a file and will update when new lines are added.

Examples:

Assume there is a file named **file.txt** which contains 20 Lines.

The following command, print last 10 lines of the file by default.

```
tail file.txt
```

```
root@tn:~# tail file.txt
This is line 11.
This is line 12.
This is line 13.
This is line 14.
This is line 15.
This is line 16.
This is line 17.
This is line 18.
This is line 19.
This is line 20.
```

The following command, print the last 5 lines of the file.

```
tail -5 file.txt
```

```
root@tn:~# tail -5 file.txt
This is line 16.
This is line 17.
This is line 18.
This is line 19.
This is line 20.
```

The following command, print the last 10 lines of the file and will append when new lines are added.

It will be helpful to view live logs for any appplications. You can close this file by **Ctrl + C** option.

```
tail -f file.txt
```

```
root@tn:~# tail -f file.txt
This is line 11.
This is line 12.
This is line 13.
This is line 14.
This is line 15.
This is line 16.
This is line 17.
This is line 18.
This is line 19.
This is line 20.
```

14. man

In Linux, the man command is used to display the documentation (or) user manual of any Linux command that can be executed on the terminal.

It includes the name of the bash command, its detailed synopsis, a short description, existing versions of the command as well as authors of the bash command.

In Linux if you don't know the syntax of the command, simply use man command to know the syntax.

Syntax:

```
man
```

```
man w
man rmdir
man pwd
man cp
```

Sections in Man command:

In the man command, the manual pages are organized into different sections to categorize and group related information. Each section corresponds to a specific type of content, and when you use the man command, you can specify the section number to retrieve information on a particular topic. The sections are denoted by numbers (typically from 1 to 9), and each section has a specific purpose.

Sections	Purpose
Section 1	Shell Commands
Section 2	System Calls
Section 3	Library Functions
Section 4	Device Files
Section 5	File Formats
Section 6	Games and Amusements
Section 7	Miscellaneous
Section 8	System Administration Tools and Deamons
Section 9	Kernel Developer's Manual

overview of some common sections:

Section 1 – User Commands

- Section 1 typically contains manual pages for user commands, which are executable programs or shell commands available to regular users.

Example:

```
man 1 pwd
```

Above command provides information about the 'pwd' command, which is an executable program.

Section 5 – File Formats

- Section 5 contains documentation about various file formats, configuration files, and conventions used on the system.
- In beginner

Example:

```
man 5 passwd
```

Above command will open the manual page for the /etc/passwd file format, which contains information about user account passwords.

15. history

The history command in Linux is used to view a history of all the commands previously executed inside the bash terminal.

Syntax:

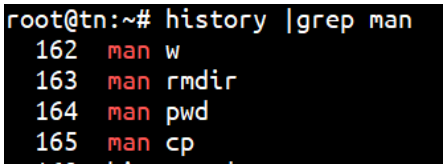
```
history
```

Command	Explanation
history | grep	You can filter any previously mentioned commands by mentioning after grep (Ex: history | grep ls)

Examples:

The following command prints how many **man** command previously executed.

```
history |grep man
```



Note: You can simply execute previous history by ! followed by history number.

For exmaple above picture 162 command is **"man w".** If again we want to execute the same command.

We can simply type **!162** it will again execute **"man w"** command.

16. clear

Clears the terminal screen. Contents will not actually be deleted in this case, only scrolled down.

You can also clear the screen by pressing **Ctrl+L** on the keyboard

Syntax:

```
clear
```

(Alternatively) Press Ctrl + L on the Keyboard.

--End of Chapter 1.

Chapter 2: Advanced Linux Commands

2.1 Linux System Information:

17. uname

The uname command is a very efficient command through which we can get all possible information about the operating system, hardware, kernel, and processor information.

Syntax:

```
uname [option]
```

Command	Explanation
uname -a	Print all informations like Kernel name, hostname, Kelnel release, Processor type, OS details

Examples:

The following example, uname -a prints all os informations.

```
root@tn:~# uname -a
Linux tn 4.4.0-142-generic #168-Ubuntu SMP Wed Jan 16 21:00:45 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
```

18. uptime

Uptime is a command that returns information about how long your system has been running together with the current time, number of users with running sessions, and the system load averages for the past 1, 5, and 15 minutes. It can also filter the information displayed at once depending on your specified options.

Syntax:

```
uptime [option]
```

```
root@tn-linux:~# uptime
 12:42:28 up 30 days,  2:34,  1 user,  load average: 0.00, 0.00, 0.00
```

Command	Explanation
uptime -p	Show uptime in detailed (Pretty) format
uptime -s	Show uptime since when

Examples:

The following example, uname -p prints total time the linux os is running.

```
uptime -p
```

```
root@tn:~# uptime -p
up 4 weeks, 1 day, 6 hours, 47 minutes
```

The following example, uname -s prints the linux os power on timestamp.

```
uptime -s
```

```
root@tn:~# uptime -s
2022-11-29 10:08:01
```

19. hostnamectl

Hostnamectl command used to view the hostname or edit the hostname of the linux machine through terminal.

Syntax:

```
hostnamectl
```

```
root@tn:~# hostnamectl
   Static hostname: tn
         Icon name: computer-vm
           Chassis: vm
        Machine ID: 7df138dee885a0a7a629b5c463858bed
           Boot ID: ecaab096a2d74b15886705b921e1a595
    Virtualization: vmware
  Operating System: Ubuntu 16.04.7 LTS
            Kernel: Linux 4.4.0-142-generic
      Architecture: x86-64
```

Command	Explanation
hostnamectl set-hostname	Rename your hostname of the linux machine

Examples:

The following example, we are changing linux machine hostname to **tn-linux** (previuosly tn).

```
hostnamectl set-hostname tn-linux
```

```
root@tn:~# hostnamectl set-hostname tn-linux
root@tn:~# hostnamectl
   Static hostname: tn-linux
         Icon name: computer-vm
           Chassis: vm
        Machine ID: 7df138dee885a0a7a629b5c463858bed
           Boot ID: ecaab096a2d74b15886705b921e1a595
    Virtualization: vmware
  Operating System: Ubuntu 16.04.7 LTS
            Kernel: Linux 4.4.0-142-generic
      Architecture: x86-64
```

20. timedatectl

The timedatectl command allows you to review and change the configuration of the system clock and its settings, you can use this command to set or change the current date, time, and timezone.

Syntax:

```
timedatectl
```

```
root@tn:~# timedatectl
      Local time: Thu 2022-12-29 12:47:04 IST
  Universal time: Thu 2022-12-29 07:17:04 UTC
        RTC time: Thu 2022-12-29 07:17:04
       Time zone: Asia/Kolkata (IST, +0530)
 Network time on: yes
NTP synchronized: yes
 RTC in local TZ: no
```

Command	Explanation
timedatectl list-timezones	List available time zones
timedatectl set-timezone	Used to set time zone
timedatectl set-time	Used to set time or date manually
timedatectl set-ntp	Used to Enable\Disable NTP. 0 for Disable & 1 for Enable.

Examples:

The following command will list all available time zones.

```
timedatectl list-timezones
```

```
root@tn:~# timedatectl list-timezones
Africa/Abidjan
Africa/Accra
Africa/Addis_Ababa
Africa/Algiers
Africa/Asmara
Africa/Bamako
Africa/Bangui
Africa/Banjul
```

The following command will set time zone to **Asia/Tokyo** (previously Asia/Kolkata).

```
timedatectl set-timezone Asia/Tokyo
```

```
root@tn:~# timedatectl set-timezone Asia/Tokyo
root@tn:~# timedatectl
      Local time: Thu 2022-12-29 16:21:23 JST
  Universal time: Thu 2022-12-29 07:21:23 UTC
        RTC time: Thu 2022-12-29 07:21:24
       Time zone: Asia/Tokyo (JST, +0900)
 Network time on: yes
NTP synchronized: yes
 RTC in local TZ: no
```

The following command will set the local time to **13:00:00**.

```
timedatectl set-time 13:00:00
```

```
root@tn:~# timedatectl set-time 13:00:00
root@tn:~# timedatectl
      Local time: Thu 2022-12-29 13:00:10 JST
  Universal time: Thu 2022-12-29 04:00:10 UTC
        RTC time: Thu 2022-12-29 04:00:10
       Time zone: Asia/Tokyo (JST, +0900)
 Network time on: no
NTP synchronized: no
 RTC in local TZ: no
```

The following command will set the local date to **2022-12-30** (previously 2022-12-29).

```
timedatectl set-time 2022-12-30
```

```
root@tn:~# timedatectl set-time 2022-12-30
root@tn:~# timedatectl
      Local time: Fri 2022-12-30 00:00:02 JST
  Universal time: Thu 2022-12-29 15:00:02 UTC
        RTC time: Thu 2022-12-29 15:00:03
       Time zone: Asia/Tokyo (JST, +0900)
 Network time on: no
NTP synchronized: no
 RTC in local TZ: no
```

The following command will enable the NTP (Network Time Protocol)

```
timedatectl set-ntp 1
```

```
root@tn:~# timedatectl set-ntp 1
root@tn:~# timedatectl
      Local time: Thu 2022-12-29 16:29:48 JST
  Universal time: Thu 2022-12-29 07:29:48 UTC
        RTC time: Thu 2022-12-29 15:04:51
       Time zone: Asia/Tokyo (JST, +0900)
 Network time on: yes
NTP synchronized: no
 RTC in local TZ: no
```

The following command will disable the NTP (Network Time Protocol)

```
timedatectl set-ntp 0
```

21. date

date command is used to display the system date and time. date command is also used to set date and time of the system.

By default, the date command displays the date in the time zone on which unix/linux operating system is configured.

You must be the super-user (root) to change date and time

Syntax:

```
date
```

```
root@tn:~# date
Thu Dec 29 14:13:57 IST 2022
```

Command	Explanation
date -d	Used to display past and future date and time
date -s	Used to set system date and time

Examples:

The following command will show what is the day 23-12-2023. Similarly you can view past and future days.

```
date -d 2023-12-29
```

```
root@tn:~# date -d 2023-12-29
Fri Dec 29 00:00:00 IST 2023
```

Note: The date command also accepts values such as "tomorrow", "Friday", "last Friday", "next Friday", "next week", and similar. For example,

```
root@tn:~# date -d yesterday
Wed Dec 28 14:39:07 IST 2022
```

The following command sets the date & time to 29-Oct-2022 15:00.

```
date -s “29 Oct 2022 15:00:00”
```

```
root@tn:~# date
Thu Dec 29 14:18:48 IST 2022
root@tn:~# date -s "29 Oct 2022 15:00:00"
Sat Oct 29 15:00:00 IST 2022
```

22. cal

cal command is a calendar command in Linux which is used to see the calendar of a specific month or a whole year in the linux terminal.

Syntax:

```
cal
```

```
root@tn:~# cal
   December 2022
Su Mo Tu We Th Fr Sa
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

Command	Explanation
cal -m	Show calendar of specified months
cal (Year)	Show calendar of specified year
cal (Month) (Year)	Show calendar of specified month of the specified year

Examples:

The following command will show the calender of first month(January). Similarly you can view Jan-Dec month calendar of current year.

```
cal -m 1
```

```
root@tn:~# cal -m 1
    January 2022
Su Mo Tu We Th Fr Sa
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
```

The following command will show the full calender of 2023 year. Similarly you can view any year calendar.

```
cal 2023
```

```
root@tn:~# cal 2023
```

The following command will show the calender of 12th month 2023 year. Similarly you can view any month & year calendar.

```
cal 12 2023
```

```
root@tn:~# cal 12 2023
   December 2023
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
```

23. w

w command in Linux is used to show who is logged on and what they are doing. This command shows the information about the users currently on the machine and their processes.

Syntax:

```
w
```

```
root@tn:~# w
 17:09:17 up 29 days,  7:01,  1 user,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
root     pts/0                     16:29    0.00s  0.06s  0.00s w
```

24. whoami

Print the username associated with the current effective user ID.

Syntax:

```
whoami
```

```
root@tn:~# whoami
root
```

2.2 Linux File Editor (Vi \ Vim)

Vi stands for **Visual**. It is a text editor that is an early attempt to a visual text editor.

Vim stands for **Vi IMproved**. It is an implementation of the Vi standard with many additions. It is the most used implementation of the standard. Most Linux distributions come with Vim already installed.

There are two major modes in **vi** & **vim** editors,

- Command Mode
- Insert Mode

By default vi or vim will open file in command mode, to edit the file we have to enter into insert mode.

Creating or Opening files using vi:

The following command opens the file named **test.txt.**

```
vim test.txt
```

Note: If **test.txt** not available then **vi** command will open the new file.

```
root@tn:~# vim test.txt
```

The following command opens the file named **test.txt** in **/tmp** directory**.**

```
vim /tmp/test.txt
```

Note: If **test.txt** not available in **/tmp** directory then **vi** command will open the new file.

```
root@tn:~# vim /tmp/test.txt
```

Editing Files using vi:

Entering into insert mode - Press **i** letter.

Escaping from insert mode - Press **Esc** button.

Saving file: - **Esc + :w**

Quit file: - **Esc + :q**

Saving file & Quit - **Esc + :wq**

Quit file without Saving - **Esc + :q!**

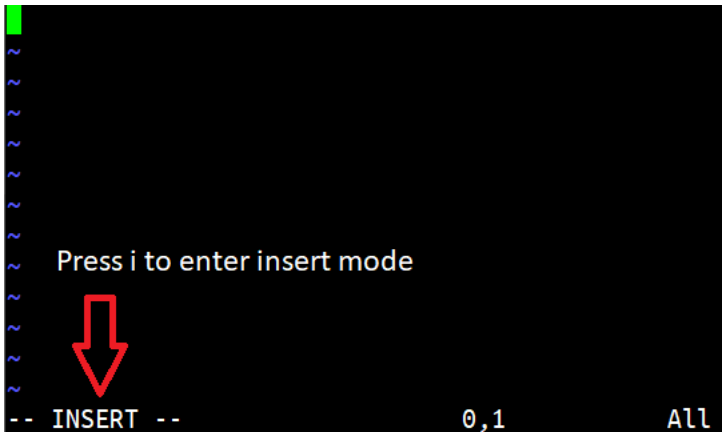
The following example, I am going to create new file called "**new.txt**" and add below lines into that file.

```
Hi Guys,
This is new file.
Edited with Vim Editor.
Thanks.
```

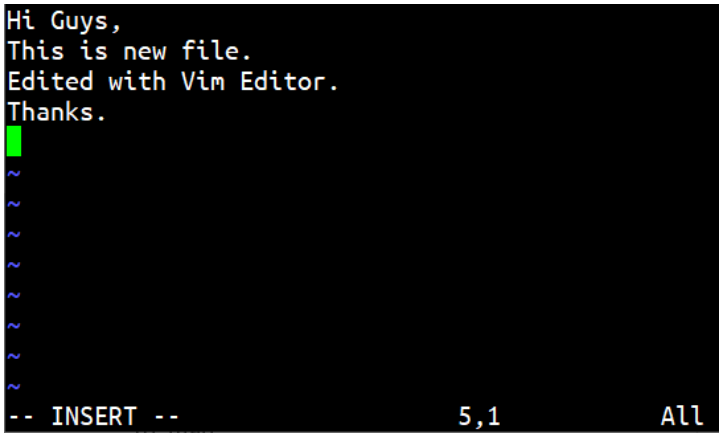
Step 1: Open **new.txt** file with **vim** editor.

```
root@tn:~# vim new.txt
```

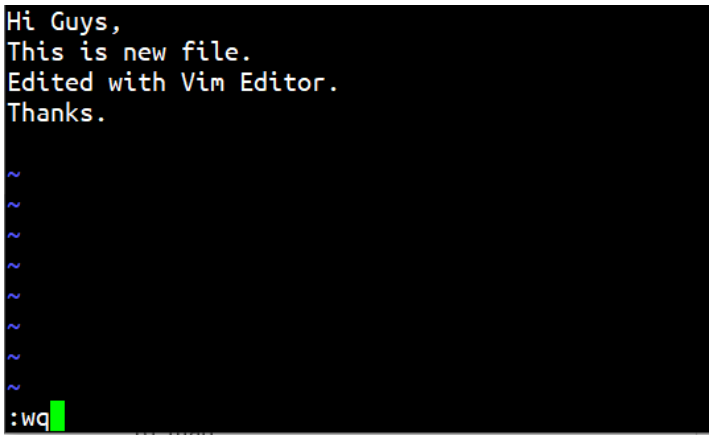
Step 2: Go to **Insert mode** by pressing the **i** button.



Step 3: Add \ Edit Lines in insert mode.

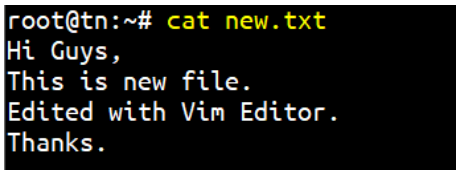


Step 4: Escape from insert mode then Save & Quit the **new.txt** file.



Step 5: Read the **new.txt** file with **cat** command.

```
cat new.txt
```



How to Copy lines in vi & vim:

To copy lines in vi or vim editors,

Go to the line which needs to be copied → Then Press **yy** → Go to the line where needs to be pasted → Then Press **p.**

Step 1: Open **new.txt** file with **vim** editor.

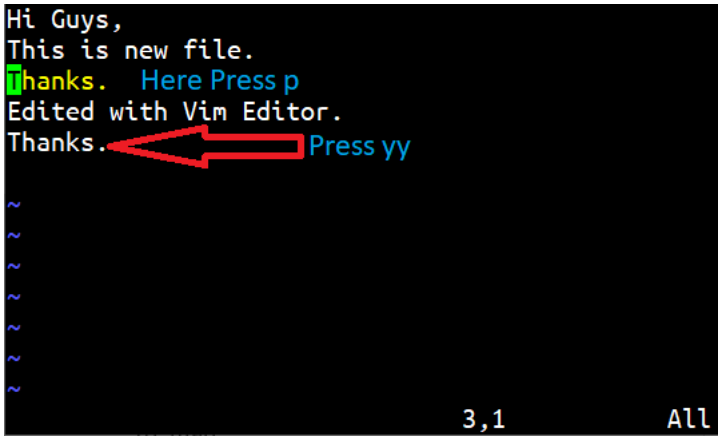
Step 2: Go to the line which need to be copied. In this example I am copying last line (Thanks).

Then Press **yy.**

```
root@tn:~# cat new.txt
Hi Guys,
This is new file.
Edited with Vim Editor.
Thanks.
```

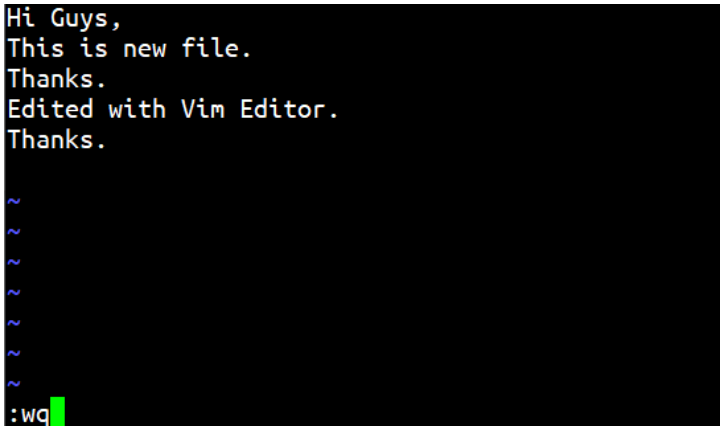
Step 3: Go to the line where the copied lines to be pasted. In this example I have pasted in line 3.

Then Press **p.**



[Note: Above steps to be performed in Command Mode (Press **Esc** then do the steps)**]**

Step 4: Save & Quit the **new.txt** file.



How to cut lines in vi & vim:

To cut lines in vi or vim editors,

Go to the line which needs to be cut → Then Press **dd** → Go to the line where needs to be pasted → Then Press **p.**

Step 1: Open **new.txt** file with **vim** editor.

```
root@tn:~# vim new.txt
```

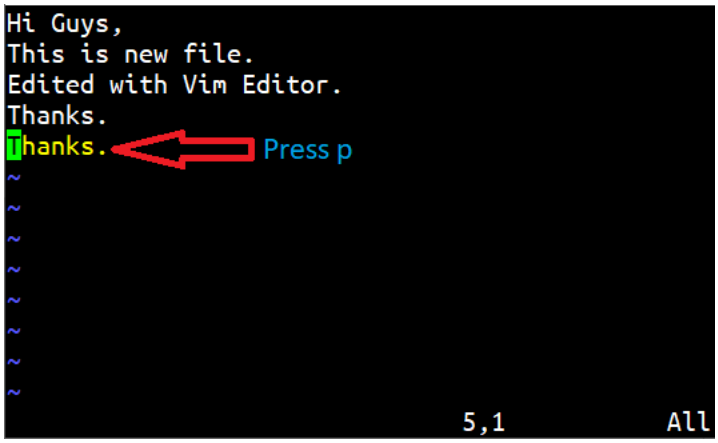
Step 2: Go to the line which need to be cut. In this example I am cutting the 3rd line (Thanks).

Then Press **dd.**

```
root@tn:~# cat new.txt
Hi Guys,
This is new file.
Thanks.
Edited with Vim Editor.
Thanks.
```

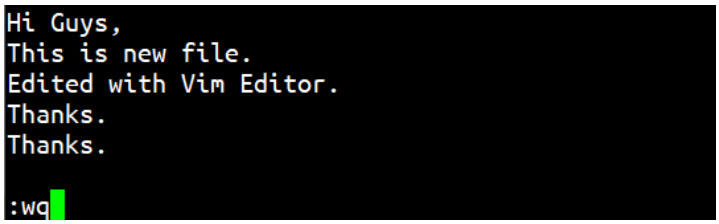
Step 3: Go to the line where the cut lines to be pasted. In this example I have pasted in last line.

Then Press **p.**



[Note: Above steps to be performed in Command Mode (Press **Esc** then do the steps)**]**

Step 4: Save & Quit the **new.txt** file.



How to delete lines in vi & vim:

To delete lines in vi or vim editors,

Go to the line which needs to be deleted → Then Press **dd.**

If you want to delete 3 lines → Then press **3dd.**

Step 1: Open **new.txt** file with **vim** editor.

```
root@tn:~# vim new.txt
```

Step 2: Go to the line which need to be delete. In this example I am cutting the last line (Thanks).

Then Press **dd.**

```
root@tn:~# cat new.txt
Hi Guys,
This is new file.
Edited with Vim Editor.
Thanks.
Thanks.
```

Step 3: Save & Quit the **new.txt** file.

```
Hi Guys,
This is new file.
Edited with Vim Editor.
Thanks.

~
~
:wq
```

[Note: Above steps to be performed in Command Mode (Press **Esc** then do the steps)**]**

How to search the word in vi & vim:

Step 1: Open **new.txt** file with **vim** editor.

```
root@tn:~# vim new.txt
```

Step 2: Type "**/searching_word"** in command mode and press enter. (Replace Searching word with the word you want to search).

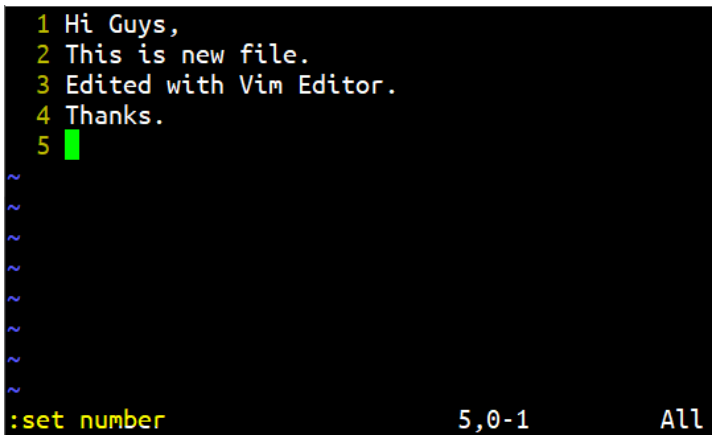
```
  1 Hi Guys,
  2 This is new file.
  3 Edited with Vim Editor.
  4 Thanks.
  5
~
~
~
~
~
~
~
~
/Edited
```

How to set the line numbers in vi & vim:

Step 1: Open **new.txt** file with **vim** editor.

```
root@tn:~# vim new.txt
```

Step 2: Type "**:set number"** in command mode and press enter.



[Note: Above steps to be performed in Command Mode (Press **Esc** then do the steps)**]**

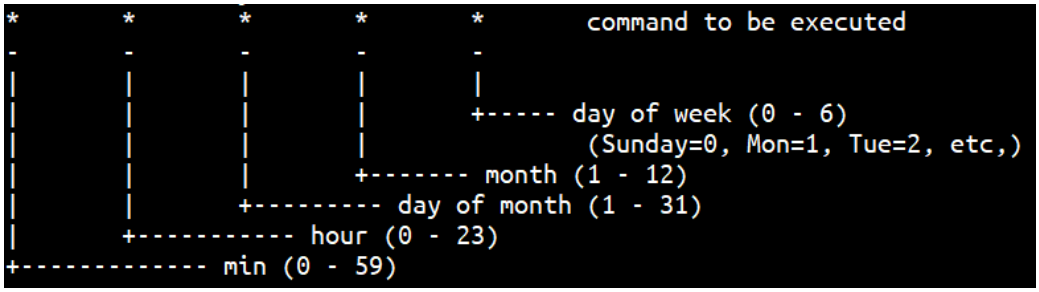
2.3 Linux Cronjob Scheduling

25. crontab

Crontab (Cron Table) is a file which contains the schedule of cron entries to be run and at specified times.

Linux Cron utility is an effective way to schedule a routine background job at a specific time and/or day on an on-going basis.

A crontab file has five fields for specifying day, date and time followed by the command to be run at that interval.



Over all 5 fields available. Each 5 fields have suitable values. For example,

- First * equals to every 1 min (0-59) **– Instead of * you can specify mins (0-59) as per your need**.
- Second * equals to every 1 hour (0-23) **– Instead of * you can specify hrs(0-23) as per your need.**
- Third * equals to every day of the month (1-31) **– Instead of * you can specify days of the month (1-31) as per your need.**
- Fourth * equals to every month (1-12) **– Instead of * you can specify month (1-12) as per your need.**
- Fifth * equals to every day of the week (0-6) - **Instead of * you can specify day of the week (0-6) as per your need.**

Syntax:

```
crontab [option]
```

Command	Explanation
crontab -e	Edit crontab file, or create one if it doesn't already exist.
crontab -l	crontab list of cronjobs , display crontab file contents.

Examples:

The following command will open the crontab file in terminal. Then we have to provide cron syntax for running.

```
crontab -e
```

Note: Find sample cron syntaxes for your reference.

Every 5 mins Cron Syntax:

The following cron syntax is for execute the script **command.sh** in background for every 5 mins.

```
*/5 * * * *    bash command.sh
```

Every 1 hour Cron Syntax:

The following cron syntax is for execute the script **command.sh** in background for every 1 hr 00 min.

```
00 */1 * * *   bash command.sh
```

Every month 1st day Cron Syntax:

The following cron syntax is for execute the script **command.sh** in background for every month 1st day 12:00 AM.

```
00 0 */1 * *   bash command.sh
```

Important: Once cron job has been scheduled, cron service must be restarted to reflect. Otherwise new cron jobs will not be executed.

The following command will restart the cron service.

```
systemctl restart crond
(Or)
Systemctl restart cron
```

Note: In some linux servers, cron service name will be **crond** & some other linux server cron service name will be **cron**.

The following command will list the scheduled cron jobs in linux server.

```
crontab -l
```

```
root@tn-linux:~# crontab -l
# Every 5 mins Cron Syntax:
*/5 * * * *     bash command.sh

# Every 1 hour Cron Syntax:
00 */1 * * *    bash command.sh

# Every month 1st day Cron Syntax:
00 0 */1 * *    bash command.sh
```

2.4 Linux Service Management - Systemd

systemd is a system and service manager for Linux operating systems. When run as first process on boot (as PID 1), it acts as init system that brings up and maintains userspace services.

26. systemctl

In the systemd utility, a service is referred to as a unit. A unit is any resource that the system knows how to act on and administrate. A unit is the principal object that the systemd tools know how to address. These assets are defined in a configuration file called a unit file.

The systemctl command is a utility which is responsible for examining and controlling the systemd system and service manager.

Syntax:

```
systemctl [subcommand] argument
```

Examples:

Start a service:

The following command starts the **mysql** service.

```
systemctl start ssh
```

```
root@tn:~# systemctl start ssh
root@tn:~#
```

Check status of a service:

The following command starts the ssh service.

```
systemctl status ssh
```

```
root@tn:~# systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; disabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-01-12 12:21:42 IST; 4min 51s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 930 (sshd)
      Tasks: 1 (limit: 2276)
     Memory: 3.9M
     CGroup: /system.slice/ssh.service
             └─930 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
```

Stop a service:

Note: If you stop a service in linux, that process will be exited and you cannot access the service. For example, if we stop the **ssh** service we cannot access this machine through ssh session from any remote machines.

And if we stopped the mysql service we cannot access mysql database.

The following command stops the **mysql** service.

```
systemctl stop mysql
```

```
root@tn:~# systemctl stop mysql
root@tn:~# systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Sun 2023-01-08 19:11:40 IST; 3 days ago
 Main PID: 84173 (code=exited, status=0/SUCCESS)

Jan 08 19:11:38 tn systemd[1]: Stopping MySQL Community Server...
Jan 08 19:11:40 tn systemd[1]: Stopped MySQL Community Server.
Jan 12 12:14:06 tn systemd[1]: Stopped MySQL Community Server.
Warning: Journal has been rotated since unit was started. Log output is incomplete or unavailable.
```

Restart a service:

A running service can be restarted using the restart command to avoid stopping and starting it manually using the following command.

The following command restarts the ssh service.

```
systemctl restart sshd
```

```
root@tn:~# systemctl restart ssh
root@tn:~#
```

Reload a service:

we do not need to restart a service to apply configuration changes, if any were you made. Instead, we can use the reload command to restart the service which implements any changes to the running service.

For example, if you made any changes in ssh service configuration file you can simply reload the ssh service without restarting the service.

The following command reload the ssh service configurations.

```
systemctl reload sshd
```

```
root@tn:~# systemctl reload ssh
root@tn:~#
```

Note: Reload is re applying the configuration for the service, but restarting is stopping the service and starting the service. So reload and restart is not same.

Enable a service:

Starting and stopping a service only applies to the current runtime. What if we need to configure the service to start when the system boots?, we have to enable the service.

The following command enable the ssh service.

```
systemctl enable ssh
```

```
root@tn:~# systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable ssh
Created symlink /etc/systemd/system/sshd.service → /lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /lib/systemd/system/ssh.service.
root@tn:~# systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-01-12 12:29:03 IST; 1min 42s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
```

Disable a service:

Likewise, if we need to configure a service to not start when the system boots, we need to disable the service.

The following command disable the ssh service.

```
systemctl disable ssh
```

```
root@tn:~# systemctl disable ssh
Synchronizing state of ssh.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable ssh
Removed /etc/systemd/system/multi-user.target.wants/ssh.service.
Removed /etc/systemd/system/sshd.service.
root@tn:~# systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; disabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-01-12 12:29:03 IST; 3min 2s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
```

2.5 Linux File Compression:

What is Compression:

compression is a reduction in the number of bits needed to represent data. Compressing data can save storage capacity, speed up file transfer and decrease costs for storage hardware and network bandwidth.

For example, if we compress 20 MB File with 2:1 ratio the compressed file size will be 10 MB approximately.

File Compression in Linux:

tar (Tape Archive), gzip are the best tools for the file compression in linux.

27. tar

The tar command in linux used to archive (compress) & extract the archived files. We archive or extract in **.gz** and **.bz2** format.

Syntax:

```
tar [option] (archive name) (file names to archive)
```

Command	Explanation
tar -c	Create archive without compression.
tar -v	Display the details of the files that have been archived.
tar -f	Creating an archive with the given file name
tar -z	Archive & compress using gzip compression(.gz).
tar -j	Archive & compress using bzip2 compression(.bz2).
tar -x	Extract the archive.
tar -t	View the files of an archive without extracting.

Examples:

The following command creates the archive file **test.tar** and add **mydoc.txt** file into the archive.

```
tar -cvf test.tar mydoc.txt
```

-c – Create archive without compression.

-v – Display the details of the files that have been archived.

-f – Create archive with the given file name (test.tar)

```
root@tn:/tmp# tar -cvf test.tar mydoc.txt
mydoc.txt
```

The following command extracts the archive file **test.tar** in current working directory.

```
tar -xvf test.tar
```

-x – Extract the archive in current working directory.

```
root@tn:/tmp# tar -xvf test.tar
mydoc.txt
root@tn:/tmp# ls mydoc.txt
mydoc.txt
```

The following command creates a tar file called **test.tar.gz** which contains the file mydoc.txt.

This command will archive & compress using gzip compression and creates the file **test.tar.gz**.

```
tar -czvf test.tar.gz mydoc.txt
```

-c – Create archive without compression.

-z – Compress using gzip compression.

-v – Display the details of the files that have been archived.

-f – Create archive with the given file name (test.tar.gz)

```
root@tn:/tmp# tar -czvf test.tar.gz mydoc.txt
mydoc.txt
root@tn:/tmp# ls test.tar.gz
test.tar.gz
```

The following command extracts the archive file **test.tar.gz** in current working directory.

```
tar -xzvf test.tar.gz
```

-x – Extract the archive in current working directory.

```
root@tn:/tmp# tar -xzvf test.tar.gz
mydoc.txt
```

The following command creates a tar file called **test.tar.bz2** which contains the file mydoc.txt.

This command will archive & compress using bzip2 compression and creates the file **test.tar.bz2**.

```
tar -cjvf test.tar.bz2 mydoc.txt
```

-c – Create archive without compression.

-j – Compress using bzip2 compression.

-v – Display the details of the files that have been archived.

-f – Create archive with the given file name (test.tar.bz2)

```
root@tn:/tmp# tar -cjvf test.tar.bz2 mydoc.txt
mydoc.txt
root@tn:/tmp# ls test.tar.bz2
test.tar.bz2
```

The following command extracts the archive file **test.tar.bz2** in current working directory.

```
tar -xjvf test.tar.bz2
```

-x – Extract the archive in current working directory.

```
root@tn:/tmp# tar -xjvf test.tar.bz2
mydoc.txt
```

The following command will list all contents of the file **test.tar.gz**.

```
tar -tf test.tar.bz2
```

-t – List the contents of an archive.

```
root@tn:/tmp# tar -tf test.tar.bz2
mydoc.txt
```

2.6 Linux Advance File Transfer Commands:

28. scp

scp is a program for copying files between computers. It uses the SSH protocol. It is included by default in most Linux and Unix distributions.

The **scp** command syntax is same like **cp** command. But the difference is cp command only copy files within the server, whereas **scp** command copy files to another machine.

Syntax:

```
scp [option] (source file) (destination file)
```

Command	Explanation
scp -r	Recursive copy for copying directories; Copies even hidden files
scp -P	Specifies the port to connect to on the remote host.
scp -p	Preserves modification times, access times & modes from the original file.

Examples:

The following example copy **file1.txt** to another linux machine **10.0.5.5 /tmp** directory.

```
scp file1.txt 10.0.5.5:/tmp/
```

```
root@tn:~# scp file1.txt 10.0.5.5:/tmp/
root@10.0.5.5's password:
file1.txt                                     100%
```

Copy directories using scp command:

The following command with -r option, will copy **mydoc** (directory) to another linux machine **10.0.5.5 /tmp/** directory.

```
scp -r mydoc/ 10.0.5.5:/tmp/
```

```
root@tn:~# scp -r mydoc 10.0.5.5:/tmp
root@10.0.5.5's password:
file1.txt                                     100%    0     0.0KB/s   00:00
```

If you didn't give **-r** option, you will get following error.

```
root@tn:~# scp mydoc 10.0.5.5:/tmp
root@10.0.5.5's password:
mydoc: not a regular file
```

SCP is using port 22 as a default port for connecting remote host. But for security reasons, your remote host changed the port into another port, You can use -P option to mention the custom ssh port.

For example,

The following command will copy **file1.txt** to another linux machine **10.0.5.5 /tmp/** directory through port no 2224.

```
scp -P 2224 file1.txt 10.0.5.5:/tmp/
```

```
root@tn:~# scp -P 2224 file1.txt 10.0.5.5:/tmp
root@10.0.5.5's password:
file1.txt                                     100%    0     0.0KB/s   00:00
```

If files & directories modification times, access times, and modes need from original files, we can preserve these from **-p** option.

For example, The following command will copy **file1.txt** to another linux machine **10.0.5.5 /tmp/** directory with preserved modification times, access times, and modes.

```
scp -p file1.txt 10.0.5.5:/tmp/
```

```
root@tn:~# ls -l file1.txt
-rw-r--r-- 1 root root 0 Jan 16 10:58 file1.txt
root@tn:~# scp -p file1.txt 10.0.5.5:/root/
root@10.0.5.5's password:
file1.txt
```

In remote machine, the file we copied will have the same timestamp details.

```
[root@redhat ~]# ls -l /root/file1.txt
-rw-r--r-- 1 root root 0 Jan 16 10:58 /root/file1.txt
```

29. rsync

Rsync, which stands for remote sync, is a remote and local file synchronization tool. It uses an algorithm to minimize the amount of data copied by only moving the portions of files that have changed.

With the help of the rsync command, you can copy and synchronize your data remotely and locally across directories, disks, and networks, perform data backups, and mirror between two Linux machines.

Syntax:

```
rsync [option] (source directory) (destination directory)
```

Command	Explanation
rsync -a	archive mode;
rsync -h	Outputs in a human readable format
rsync -v	information about what files are being transferred
rsync -z	compress file data during the transfer
rsync --ignore-existing	skip updating files that already exist on destination
rsync --delete	delete destination files if not available in source location

Examples:

Transfer / Sync a Directory on Local Machine:

The following command will transfer or sync all the files from one directory to a different directory in the same machine.

For Example, The following command will transfer or sync files from **mydoc/** directory to **/tmp/rsync/** directory.

```
rsync -avzh mydoc/ /tmp/rsync/
```

-a – Archive mode (preserves permissions & timestamp details),

-v – Displays the information about transferred files,

-z – Compress file data during the transfer,

-h – Displays the transferred file size details in human readable format.

```
root@tn:~# rsync -avzh mydoc/ /tmp/rsyc/
sending incremental file list
created directory /tmp/rsyc
./
capture1.pcap
file1.txt

sent 965 bytes  received 89 bytes  2.11K bytes/sec
total size is 9.55K  speedup is 9.06
```

- Here **mydocs/** is a source directory,
- **/tmp** is a destination directory.

As you can see above image rsync will automatically create the **rsync** directory inside **/tmp** directory.

[**Note:** If destination directory not available rsync command will create the directory then transfers the files.]

Transfer / Sync a Directory from Local Machine to Remote Machine:

The following command will transfer or sync all the files from one linux machine directory to another remote machine directory.

```
rsync -avzh mydoc/ 10.0.5.5:/tmp/rsync/
```

```
root@tn:~# rsync -avzh mydoc/ 10.0.5.5:/tmp/rsync/
root@10.0.5.5's password:
sending incremental file list
created directory /tmp/rsync
./
capture1.pcap
file1.txt

sent 965 bytes  received 90 bytes  162.31 bytes/sec
total size is 9.55K  speedup is 9.05
```

Transfer / Sync a Directory from Local Machine to Remote Machine & ignore existing files:

The following command will transfer or sync all the files from one linux machine directory to another remote machine directory. And this will not transfer or update files that are already existing in destination.

```
rsync -avzh --ignore-existing mydoc/ 10.0.5.5:/tmp/rsync/
```

-a – Archive mode (preserves permissions & timestamp details),

-v – Displays the information about transferred files,

-z – Compress file data during the transfer,

-h – Displays the transferred file size details in human readable format,

–-ignore-existing – Don't Copy or Sync already existing files in destination.

```
root@tn:~# rsync -avzh --ignore-existing mydoc/ 10.0.5.5:/tmp/rsync/
root@10.0.5.5's password:
sending incremental file list
./
new.txt

sent 256 bytes  received 38 bytes  65.33 bytes/sec
total size is 9.62K  speedup is 32.71
root@tn:~# ls mydoc/
capture1.pcap  file1.txt  new.txt
```

In Above example, rsync command transferred **new.txt** file only to the destination directory (**10.0.5.5:/tmp/rsync**). Because other files has been already transferred to destination directory with previous example.

Transfer / Sync a Directory from Local Machine to Remote Machine with -delete Option:

Example Scenario,

- If you want to transfer or sync files from **mydos/** directory to **/tmp/rsync** directory.
- But in **/tmp/rsync** directory, already **test.txt** file is available which is not available in source directory **mydocs.**
- Now, you want only files that are available in **mydocs** directory to sync with **/tmp/rsync** directory & don't want **test.txt** file to be available in destination directory **/tmp/rsync**.

We can use the **'–delete'** option to delete files that are not there in the source directory.

The following command will transfer or sync all the files from one directory to another directory. And -delete option will remove files that are not available in source directory.

```
rsync --delete -avzh mydoc/ 10.0.5.5:/tmp/rsync/
```

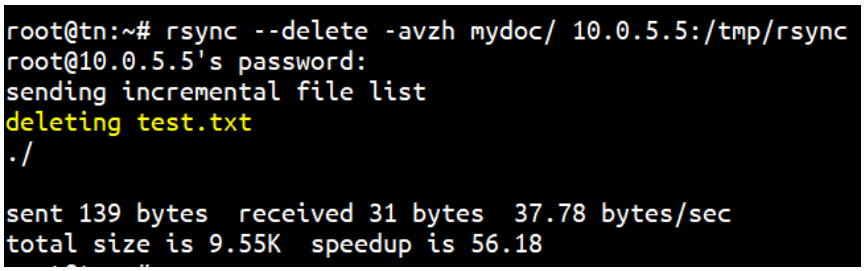
-a – Archive mode (preserves permissions & timestamp details),

-v – Displays the information about transferred files,

-z – Compress file data during the transfer,

-h – Displays the transferred file size details in human readable format,

–-delete – Delete existing files in destination location, that are not available in Source location.



Transfer / Sync a Directory from Local Machine to Remote Machine using SSH:

The following command will transfer or sync all the files from one linux machine directory to another remote machine directory in a secured way with encryption using ssh protocol.

To specify a ssh protocol you need to give -e option along with rsync other options.

```
rsync -avzhe ssh mydoc/ 10.0.5.5:/tmp/rsync/
```

-a – Archive mode (preserves permissions & timestamp details),

-v – Displays the information about transferred files,

-z – Compress file data during the transfer,

-h – Displays the transferred file size details in human readable format,

-e ssh – Use SSH Protocol to transfer files to remote host.

```
root@tn:~# rsync -avzhe ssh mydoc/ 10.0.5.5:/tmp/rsync/
root@10.0.5.5's password:
sending incremental file list
./
ssh-test.txt

sent 204 bytes  received 38 bytes  19.36 bytes/sec
total size is 9.55K  speedup is 39.46
```

--End of Chapter 2.

Chapter 3: Linux Performance Monitoring & Statistics:

30. free

The Linux free command outputs a summary of RAM usage, including total, used, free, shared, and available memory and swap space. The command helps monitor resource usage and allows an admin to determine if there's enough room for running new programs.

free gathers information by parsing the /proc/meminfo file.

Syntax:

```
free [option]
```

Command	Explanation
free -h	Display amount of free and used memory in human readable format

Examples:

The following command will show the free & used memory details in human readable format.

```
root@tn:~# free -h
              total        used        free      shared  buff/cache   available
Mem:           7.8G        367M        2.0G         24M        5.4G        7.1G
Swap:          979M        884K        979M
```

31. df

The df (disk free) command is used to display total space and available space of the file system.

Syntax:

```
df [option]
```

Command	Explanation
df -h	Display disk space available on the file system in human readable format

Examples:

The following command list disk usage details in human readable format.

```
root@tn:~# df -h
Filesystem                  Size  Used Avail Use% Mounted on
udev                        3.9G     0  3.9G   0% /dev
tmpfs                       797M   17M  780M   3% /run
/dev/mapper/mysql--vg-root   73G   26G   44G  37% /
tmpfs                       3.9G     0  3.9G   0% /dev/shm
tmpfs                       5.0M     0  5.0M   0% /run/lock
tmpfs                       3.9G     0  3.9G   0% /sys/fs/cgroup
/dev/sda1                   720M  111M  573M  17% /boot
```

32. du

The “disk usage” command in used to view files & directories utilized space in hard disk. The du command can be used to track the files and directories which are consuming excessive amount of space on hard disk drive.

Syntax:

```
du [option]
```

Command	Explanation
du -h	Print content sizes in human readable format(K, M, G)
du -sh	Display total disk usage size in human readable format

Examples:

The following command shows disk usage of files & directories available in **current working** directory.

```
du -h
```

```
root@tn:~# du -h
8.0K    ./mydocs/mydoc1
16K     ./mydocs
8.0K    ./.ssh
4.0K    ./document/tech
8.0K    ./document
4.0K    ./.cache
100K    .
```

The following command shows disk usage of files & directories available in **mydocs/** directory.

```
du -h mydocs/
```

```
root@tn:~# du -h mydocs/
8.0K    mydocs/mydoc1
16K     mydocs/
```

The following command shows disk usage & total size of files & directories available in **mydocs/** directory.

```
du -ch mydocs/
```

```
root@tn:~# du -ch mydocs/
8.0K    mydocs/mydoc1
16K     mydocs/
16K     total
```

The following command shows summary (it wont show inside directory contents) of disk usage & total size of files & directories available in **mydocs/** directory.

```
du -sh
```

```
root@tn:~# du -sh
8.0K    document
4.0K    file.txt
16K     mydocs
4.0K    test.txt
4.0K    ty
```

33. top

The top utility is a commonly used tool for displaying system-performance information. It dynamically shows administrators which processes are consuming processor and memory resources.

Usually, this command shows the summary information of the system and the list of processes or threads which are currently managed by the Linux Kernel.

top command will open an interactive command mode where the top half portion will contain the statistics of processes and resource usage. And Lower half contains a list of the currently running processes. Pressing q will exit the command.

- **top** displays uptime information
- **Tasks** displays process status information
- **%Cpu(s)** displays various processor values
- **MiB Mem** displays physical memory utilization
- **MiB Swap** displays virtual memory utilization

Syntax:

```
top
```

```
top - 16:56:41 up 30 days,  6:48,  1 user,  load average: 0.00, 0.00, 0.00
Tasks: 179 total,   1 running, 178 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.3 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  8156840 total,  2076816 free,   371152 used,  5708872 buff/cache
KiB Swap:  1003516 total,  1002632 free,      884 used.  7403944 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
  748 root      20   0  190104   9072   7832 S   0.3  0.1  18:45.75 vmtoolsd
    1 root      20   0  185240   5820   3976 S   0.0  0.1   0:21.15 systemd
```

Top Command will list following details:

```
PID     : Shows task's unique process id.
USER    : Username of owner of task.
PR      : The process's priority. The lower the number, the higher the priority.
NI      : Represents a Nice Value of task.Negative nice value implies higher priority & positive Nice value means lower priority.
VIRT    : Total virtual memory used by the task.
RES     : How much physical RAM the process is using, measured in kilobytes.
SHR     : Represents the Shared Memory size (kb) used by a task.
S       : This field shows the process state in the single-letter form
%CPU    : Represents the CPU usage.
TIME+   : CPU Time, the same as 'TIME', but reflecting more granularity through hundredths of a second.
%MEM    : Shows the Memory usage of task.
COMMAND : The name of the command that started the process.
```

Examples:

Command	Explanation
top -u	Display only processes with a user id or user name matching that given

--End of Chapter 3.

Chapter 4: Linux Process Management:

34. ps

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 (process status): displays information about processes.

Syntax:

```
ps [option]
```

Command	Explanation
ps aux	Display every process on BSD Format.
ps -ef	Display every process on standard format(Commonly Used).

Examples:

```
ps aux
```

```
root@tn:~# ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.0 185240  5820 ?        Ss   Nov29   0:21 /lib/systemd/systemd
root           2  0.0  0.0      0     0 ?        S    Nov29   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        S    Nov29   0:00 [ksoftirqd/0]
root           5  0.0  0.0      0     0 ?        S<   Nov29   0:00 [kworker/0:0H]
root           7  0.0  0.0      0     0 ?        S    Nov29   2:48 [rcu_sched]
```

The standard format for viewing the process.

```
ps -ef
```

```
root@tn:~# ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 Nov29 ?        00:00:21 /lib/systemd/systemd
root           2       0  0 Nov29 ?        00:00:00 [kthreadd]
root           3       2  0 Nov29 ?        00:00:00 [ksoftirqd/0]
root           5       2  0 Nov29 ?        00:00:00 [kworker/0:0H]
root           7       2  0 Nov29 ?        00:02:48 [rcu_sched]
```

We can combine the **ps** command with **grep** command to view the specific process.

The following command shows the process details of ssh.

```
ps -ef |grep ssh
```

```
root@tn:~# ps -ef |grep ssh
root         850   43424  0 15:26 ?        00:00:00 sshd: root@pts/0
root        2570     939  0 17:24 pts/0    00:00:00 grep --color=auto ssh
root       43424       1  0  2022 ?        00:00:00 /usr/sbin/sshd -D
```

35. kill

kill command is used to terminate processes manually.

It sends a signal which ultimately terminates or kills a particular process or group of processes.

If you want to stop a process, specify the process ID (PID) in the ProcessID variable.

Syntax:

```
kill [option] (pid)
```

Command	Explanation
kill	Kill the process of mentioned Process ID.
kill -9	Forcefully Kill the process of mentioned Process ID.
pkill	Kill the process of mentioned Process Name.

Example:

The following command will kill the process with PID 12002.

```
kill 12002
```

```
root@tn:~# kill 12002
```

The following command will forcefully kill the process with PID 12002. Use this option if normal kill not working.

```
kill -9 12002
```

```
root@tn:~# kill -9 12002
```

The following command will kill the process with the name **free**.

```
pkill free
```

--End of Chapter 4.

Chapter 5: Linux Users and Group Management:

Three types of Linux Accounts:

Root account – Default Administrator Account

User accounts – Normal Users

System accounts – created only for a specific purpose or software. For example, a mail/apache accounts.

Understanding Passwd, Shadow Files:

/etc/passwd – Keeps the user account name, acc group, acc description and password information.

This file holds most of the information about accounts on the Unix system.

/etc/shadow – Holds the encrypted password of the corresponding account.

/etc/group – This file contains the group information for each account.

36. useradd

useradd command used to create new users in linux operating system.

Useradd command performs the following major things:

- It edits /etc/passwd, /etc/shadow and /etc/group and files for the newly created user accounts.
- Creates a home directory for the new user.
- Sets permissions and ownerships to the home directory.

All new, existing & system usernames & details are available in **/etc/passwd** file.

Syntax:

```
useradd [options] (New user id)
```

Command	Explanation
useradd -m	Creates new user with home directory in /home/
useradd -d	Creates new user with custom home directory.
useradd -g	Creates new user & add mentioned group name or ID as primary group.
useradd -s	Specify the custom shell.
useradd -e	The date (format YYYY-MM-DD) on which the user account will be disabled.

Example:

The following command creates the user **trucks** with default home directory **(/home/trucks).**

```
useradd -m trucks
```

```
root@tn:~# useradd -m trucks
```

The following command creates the user **trucks** with custom home directory **(/trucks-user).**

```
useradd -d /truck-user trucks
```

```
root@tn:~# useradd -d /truck-user trucks
```

The following command creates the user **trucks** and add development group as primary group. by default primary group will be created same name as user name.

```
useradd -g development trucks
```

```
root@tn:~# useradd -g development trucks
```

The following command creates the user **trucks** and sets the default shell for the user to **/bin/zsh** shell.

```
useradd -s /bin/zsh trucks
```

```
root@tn:~# useradd -s /bin/zsh trucks
root@tn:~# cat /etc/passwd |grep zsh
trucks:x:1003:1004::/home/trucks:/bin/zsh
```

The following command creates the user **trucks** and sets user expiry date to **2023-12-31.**

```
useradd -e 2023-12-31 trucks
```

```
root@tn:~# useradd -e 2023-12-31 trucks
root@tn:~# chage -l trucks
Last password change                                    : Dec 30, 2022
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : Dec 31, 2023
```

Combination of Multiple Options:

```
useradd -m -s /bin/bash -c "Trucks User" -e 2023-12-31 -g development trucks
```

```
root@tn:~# useradd -m -s /bin/bash -c "Trucks User" -e 2023-12-31 -g development trucks
```

Above command creates user **trucks** with following option,

-m : Creates the default home directory **/home/trucks**.

-s : Making **/bin/bash** as default shell.

-c : Adding description as **"Trucks User".**

-e : Sets expiry date as **2023-12-31**.

-g : Making **development** group as primary group.

37. usermod

The usermod command used to modify existing linux users settings & parameters. Most of the useradd command options are available with same functionality in usermod command.

Syntax:

```
usermod [options]
```

Command	Explanation
usermod -c	Add / Modify User description .
usermod -d	New home directory for the user account.
usermod -e	Set expiry date for the user account.
usermod -g	Modify Primary Group.
usermod -aG	Append the user to the additional group without removing from other groups.
usermod -l	Modify User Login Name
usermod -m	Move user home directory to other location.
usermod -L	Lock User.
usermod -U	Unlocak User.
usermod -s	Change user default shell.

Example:

The following command modify the user description to **"User Trucks"**.

```
usermod -c "User Trucks" trucks
```

```
root@tn:~# usermod -c "User Trucks" trucks
```

The following command modify the user home directory to **"/trucks-user"**.

```
usermod -d /trucks-user trucks
```

```
root@tn:~# usermod -d /trucks-user trucks
root@tn:~# cat /etc/passwd|grep trucks
trucks:x:1003:1004:User Trucks:/trucks-user:/bin/bash
```

The following command modify the user home directory to **"/home/trucks1"** and moves the files available in older home directory to new home directory.

```
usermod -d /home/trucks1 -m trucks
```

```
root@tn:~# usermod -d /home/trucks1 -m trucks
```

The following command modify the user account expiry date to **"2023-06-01"**.

```
usermod -e 2023-06-01
```

```
root@tn:~# usermod -e 2023-06-01 trucks
root@tn:~# chage -l trucks
Last password change                                    : Dec 30, 2022
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : Jun 01, 2023
```

The following command modify the user primary group to **"dev-team"**.

```
usermod -g dev-team trucks
```

```
root@tn:~# usermod -g dev-team trucks
```

The following command add the user to additional group **"development"**.

```
usermod -aG development trucks
```

```
root@tn:~# usermod -aG development trucks
root@tn:~# id trucks
uid=1003(trucks) gid=1004(dev-team) groups=1004(dev-team),1002(development)
```

The following command modify the user login name from trucks to **"trucks-usr"**.

```
usermod -l trucks-usr trucks
```

```
root@tn:~# usermod -l trucks-usr trucks
usermod: user trucks is currently used by process 107723
root@tn:~# usermod -l trucks-usr trucks
```

Note: Sometimes while modifying user login it will throw error if any of the modifying user process running background. If does kill the process and again execute the command.

The following command locks the user account.

```
usermod -L trucks-usr
```

```
root@tn:~# usermod -L trucks-usr
```

The following command unlocks the user account.

```
usermod -U trucks-usr
```

```
root@tn:~# usermod -U trucks-usr
usermod: unlocking the user's password would result in a passwordless account.
You should set a password with usermod -p to unlock this user's password.
```

Note: Set the new password to unlocked account.

The following command modify the default shell from **/bin/bash** to **/bin/zsh**.

```
usermod -s /bin/zsh trucks
```

38. userdel

The userdel command used to delete the user account with or without user home directory.

Syntax:

```
Syntax for Delete without user home directory & files:
userdel [User Name]

Syntax for Delete with user home directory & files:
userdel -r [User Name]
```

Example:

The following command deletes the user trucks but not his home directory. So userdel command without any options will delete the user and removes the entry in /etc/passwd file, but its not deleting the user home directoty.

```
userdel trucks
```

```
root@tn:~# userdel trucks
root@tn:~# cat /etc/passwd |grep trucks
root@tn:~# ls /home |grep truck
trucks
```

If you want to delete user with his home directory and files, run userdel command with **-r** option.

```
userdel -r trucks
```

```
root@tn:~# userdel -r trucks
userdel: trucks mail spool (/var/mail/trucks) not found
root@tn:~# cat /etc/passwd |grep trucks
root@tn:~# ls /home |grep truck
root@tn:~#
```

39. passwd

The passwd command in linux used change the user password.

Syntax:

```
passwd
```

Note: Root user can change any user password using passwd followed by username.

Example:

The following command used to change password of root (currently logged in user).

```
passwd
```

```
root@tn:~# passwd
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

The following command used to change password of other user by root.

```
Passwd trucks
```

```
root@tn:~# passwd trucks
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

40. chage

The chage command is used to view and change the user password expiry information. This command is used when the login is to be provided for a user for a limited amount of time or when it is necessary to change the login password from time to time.

Syntax:

```
chage [options]
```

Command	Explanation
chage -l	Used to view the account aging information
chage -E	Used to set the account expiry.
chage -m	Used to set the minimum number of days between password changes.
chage -M	Used to set maximum days for password validity.(Passwrd Expiry date)
chage -I	Used to set account inactive after password expiry.
chage -W	Gives prior warning before the password expiry.

Example:

The following command used to view the aging information like last password change date, password expiry date, account inactive days, account expiry date & password change warning days of account **trucks.**

```
chage -l trucks
```

```
root@tn:~# chage -l trucks
Last password change                                    : Jan 08, 2023
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : never
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
```

The following command used you to change the last password change date to a new date **2023-01-30** for the account **trucks.**

```
chage -d 2023-01-30 trucks
```

```
root@tn:~# chage -d 2023-01-31 trucks
root@tn:~# chage -l trucks
Last password change                                    : Jan 31, 2023
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 1
```

The following command used to set account expiry day to **2023-02-05** of account **trucks.**

```
chage -E 2023-02-05 trucks
```

```
root@tn:~# chage -E 2023-02-05 trucks
root@tn:~# chage -l trucks
Last password change                                    : Jan 31, 2023
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : Feb 05, 2023
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 1
```

As you can see, now its showing account expiry date as 02-Feb-2023 for the **trucks** user.

The following command used to set minimum days between password change. If trucks user changed his password today, then he cannot change his password for next two days.

The following sets **trucks** user cannot change his password minimum 2 days after changed the account password.

```
chage -m 2 trucks
```

```
root@tn:~# chage -m 2 trucks
root@tn:~# chage -l trucks
Last password change                                    : Jan 31, 2023
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : Feb 05, 2023
Minimum number of days between password change          : 2
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 1
```

The following command used to set maximum password valid date to **5 days** for **trucks** user.

after 5 days of account password expired.

And also this will calculate the password with last password change. For example,

If Last password change date is **31-Jan-2023**, and if you set the Maximum password validity to 5 days then the calculation will be ,

31-Jan-2023 + 5 days = 05-Feb-2023 (**trucks** account password will expire by **05-Feb-2023**).

```
chage -M 5 trucks
```

```
root@tn:~# chage -M 5 trucks
root@tn:~# chage -l trucks
Last password change                                    : Jan 31, 2023
Password expires                                        : Feb 05, 2023
Password inactive                                       : never
Account expires                                         : Feb 05, 2023
Minimum number of days between password change          : 2
Maximum number of days between password change          : 5
Number of days of warning before password expires       : 1
```

The following command used to set 5 password inactive days after **trucks** user password expired. Means **trucks** user can change his password after 5 days of account password expired. For example,

If password expiry date is **05-Feb-2023**, and if you set the password inactive days to 5 days then the calculation will be ,

05-Feb-2023 + 5 days = 10-Feb-2023 (**trucks** account can change his password after 5 days of his password expiry also).

```
chage -I 5 trucks
```

```
root@tn:~# chage -I 5 trucks
root@tn:~# chage -l trucks
Last password change                                    : Jan 31, 2023
Password expires                                        : Feb 05, 2023
Password inactive                                       : Feb 10, 2023
Account expires                                         : Feb 05, 2023
Minimum number of days between password change          : 2
Maximum number of days between password change          : 5
Number of days of warning before password expires       : 1
```

The following command used to set warning before 3 days of password expiry for the account **trucks.**

```
chage -w 3 trucks
```

```
root@tn:~# chage -W 3 trucks
root@tn:~# chage -l trucks
Last password change                                    : Jan 31, 2023
Password expires                                        : Feb 05, 2023
Password inactive                                       : Feb 10, 2023
Account expires                                         : Feb 05, 2023
Minimum number of days between password change          : 2
Maximum number of days between password change          : 5
Number of days of warning before password expires       : 3
```

41. groupadd

The groupadd command used to create new group in linux with specified group name.

All new, existing & system group names & details are available in **/etc/group** file.

Syntax:

```
groupadd [new group name]
```

Command	Explanation
groupadd -g	Creates new group with custom group id (GID).

Example:

The following command creates the new group called **development**.

```
groupadd development
```

```
root@tn:~# groupadd development
```

The following creates the group **dev-group** with 7500 (custom) group id.

```
groupadd -g 7500 dev-group
```

```
root@tn:~# groupadd -g 7500 dev-group
root@tn:~# cat /etc/group |grep dev-group
dev-group:x:7500:
```

42. groupdel

The groupdel command used to delete the existing group in linux operating systems.

Syntax:

```
groupdel [existing group name]
```

Example:

The following deletes the group called **dev-group**.

```
groupdel dev-group
```

```
root@tn:~# groupdel dev-group
root@tn:~# cat /etc/group |grep dev-group
root@tn:~#
```

43. groupmod

The groupmod command used to modify the existing group name.

Syntax:

```
groupmod [new group name] old-group-name
```

Example:

The following command changes the group name **development** to **IT-development**.

```
groupmod -n IT-development development
```

```
root@tn:~# groupmod -n IT-development development
root@tn:~# cat /etc/group |grep IT
IT-development:x:1002:
```

5.1 sudo & visudo Command in Linux:

Sudo Command:

What is the need for sudo?

Giving root user password to all admins is dangerous for many reasons.

Working as root means that you have the power to:

- Remove any or all files
- Change the permissions of any or all files
- Change the runlevel of the system
- Alter user accounts

- Mount or unmount filesystems
- Remove or install software
- Create, remove, and alter file systems

Basically, you can do anything to the system as the root user. It is the all-powerful administrative account. And, unlike other operating systems, you won't see a, "Are you sure?" dialog to be sure that the rm -rf * command you just issued was in /opt/tmp rather than at /. As you can imagine, errors made as the root user can be irreversible and devastating.

To overcome these problems, we can use sudo privileges.

44. sudo

sudo is a command that runs an elevated prompt without a need to change your identity. Depending on your settings in the /etc/sudoers file, you can issue single command as root or as another user.

For example, if you want to update the packages, you run:

```
apt-get update
```

But you will see an error if you are not logged in as a root user.

```
sysadmin@tn:~$ apt-get update
Reading package lists... Done
W: chmod 0700 of directory /var/lib/apt/lists/partial failed - SetupAPTPartialDirectory (1: Operation not permitted)
E: Could not open lock file /var/lib/apt/lists/lock - open (13: Permission denied)
E: Unable to lock directory /var/lib/apt/lists/
W: Problem unlinking the file /var/cache/apt/pkgcache.bin - RemoveCaches (13: Permission denied)
W: Problem unlinking the file /var/cache/apt/srcpkgcache.bin - RemoveCaches (13: Permission denied)
```

Instead, if you run the same command with sudo:

```
sudo apt-get update
```

You will be asked to type your password, and then you can run the command if you are a part of the sudo group.

```
sysadmin@tn:~$ sudo apt-get update
sudo: unable to resolve host tn: Connection timed out
[sudo] password for sysadmin:
Get:1 http://security.ubuntu.com/ubuntu xenial-security InRelease [99.8 kB]
Hit:2 http://us.archive.ubuntu.com/ubuntu xenial InRelease
Get:3 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease [99.8 kB]
Get:4 https://esm.ubuntu.com/infra/ubuntu xenial-infra-security InRelease [7,524 B]
Get:5 https://esm.ubuntu.com/infra/ubuntu xenial-infra-updates InRelease [7,475 B]
Get:6 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease [97.4 kB]
Fetched 312 kB in 2s (139 kB/s)
Reading package lists... Done
sysadmin@tn:~$
```

How to give sudo privilege to the user?

By adding a user and required permission details to /etc/sudoers file, we can give sudo privilege to any user.

45. visudo

The visudo command is a safe and secure way of editing the /etc/sudoers file on Linux OS.

Visudo locks the sudoers file against multiple simultaneous edits, provides basic sanity checks, and checks for parse errors.

If the sudoers file is currently being edited by someone else, or by you in another session, you will receive a message to try again later.

Syntax:

```
visudo → Press Enter
```

```
root@tn:~# visudo
```

5.2 How to give full access to a User account:

Example:

The following example we are providing full access to the **trucks** user in linux machine.

```
visudo → Press Enter

## Add below lines in the end of the file.
trucks ALL=(ALL:ALL)  ALL

## Now save the file with following steps in vi editor,
Press ESC Key → :wq
```

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
trucks  ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
~
:wq
```

User trucks can run all commands like root user with sudo option.

```
root@tn:~# su - trucks
trucks@tn:~$ sudo -l
[sudo] password for trucks:
Matching Defaults entries for trucks on tn:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User trucks may run the following commands on tn:
    (ALL : ALL) ALL
trucks@tn:~$
```

5.3 How to give particular command execution access to a User account:

The following example, we are providing **apt-get** command execution only access to **trucks** user.

So trucks user cannot execute any commands other than apt-get command.

```
visudo → Press Enter
## Add below lines in the end of the file.

trucks ALL=(ALL:ALL) /usr/bin/apt-get

## Now save the file with following steps in vi editor,
Press ESC Key → :wq
```

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
trucks  ALL=(ALL:ALL) /usr/bin/apt-get

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
~
:wq
```

Note: Its important to mention the binary path of the command which you want give sudo access to the user.

If you don't know the correct binary path of the command use which command to get the path,

For example, to know the **apt-get command binary path**, run following command

```
which apt-get
```

```
root@tn:~# which apt-get
/usr/bin/apt-get
```

How to check allocated privileges for a user?

The following command will list the list of given sudo access to the trucks (currently logged in) user.

```
sudo -l
```

```
trucks@tn:~$ sudo -l
Matching Defaults entries for trucks on tn:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User trucks may run the following commands on tn:
    (ALL : ALL) /usr/bin/apt-get
```

User trucks cannot run other than given sudo permission.

```
trucks@tn:~$ sudo apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu xenial InRelease
Hit:2 http://security.ubuntu.com/ubuntu xenial-security InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease
Hit:5 https://esm.ubuntu.com/infra/ubuntu xenial-infra-security InRelease
Hit:6 https://esm.ubuntu.com/infra/ubuntu xenial-infra-updates InRelease
Reading package lists... Done
trucks@tn:~$
trucks@tn:~$ sudo useradd -c "Test 2 User" test2
Sorry, user trucks is not allowed to execute '/usr/sbin/useradd -c Test 2 User test2' as root on tn.
```

--End of Chapter 5.

Chapter 6: Linux Permission Management:

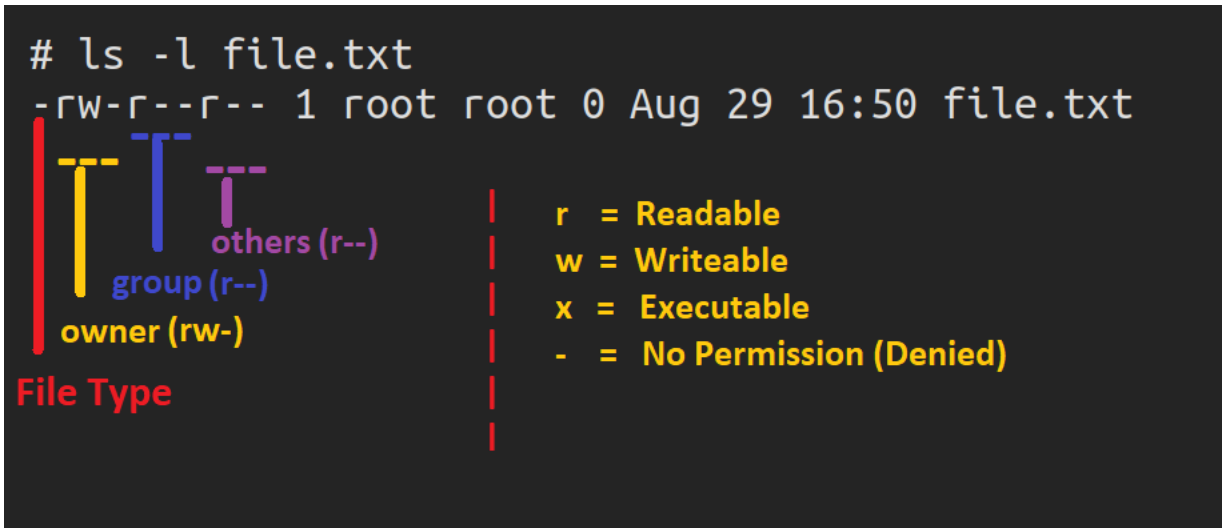
Permissions in Linux:

There are three classes of permissions for any file, directory in linux.

- User
- Group
- Other

There are three permissions for any file, directory in linux.

- r — Indicates that a given category of user can read a file.
- w — Indicates that a given category of user can write to a file.
- x — Indicates that a given category of user can execute the file.



- 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 File \ directory Permissions:

There are 2 ways to use the command:

1) Symbolic mode

2) Absolute mode

Symbolic mode:

In the symbolic mode, you can modify permissions of a specific owner. It makes use of mathematical symbols to modify the file 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

If you run ls -l command in linux, you can see symbolic mode permission for file & directories in left hand side.

Operator	Description
+	Adds a permission to a file or directory
-	Removes the permission
=	Sets the permission and overrides the permissions set earlier.

The various owners are represented as -

User Denotations	
u	user/owner
g	group
o	other
a	all

Absolute mode:

- Absolute (Numeric) Mode
- In this mode, file permissions are represented as a three-digit octal number.
- **Read – 4, Write – 2, Execute – 1** –> 3 important Permissions numbers.

So what number would you use if you wanted to set a permission to read and write? 4 + 2 = 6.

Likewise some sample permissions for absolute & symbolic mode.

Symbolic Mode	Absolute Mode	Given Permission
-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
--wx-wx-wx	0333	All classes can write/execute
-r--r--r--	0444	All classes can read
--w--w--w-	0222	All classes can write
---x--x--x	0111	All classes can execute
-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
----------	0000	None of the classes have permissions

46. chmod

The chmod command is used to change the permissions of a file or directory.

Syntax:

```
chmod [option] [permission] filename
```

Command	Explanation
chmod -R	Change files and directories permission recursively

Example:

Set / Remove Permission with Symbolic Mode:

The following command sets the Read, Write, Execute permission to User (Owner) for the file file1.txt.

```
chmod u+rwx file1.txt
```

```
root@tn:/tmp# ls -l file1.txt
-rwxr--r-- 1 root root 32 Dec 27 14:08 file1.txt
```

The following command set the Read, Write, Execute permission to User, Group & Others Classes for the folder1 directory.

```
chmod a+rwx folder1
```

```
root@tn:/tmp# chmod a+rwx folder1
root@tn:/tmp# ls -l |grep folder1
drwxrwxrwx 2 root root 4096 Dec  3 19:45 folder1
```

The following command removes the Write & Execute permission to Group Class for the folder1 directory.

```
chmod g-wx folder1
```

```
root@tn:/tmp# chmod g-wx folder1
root@tn:/tmp# ls -l |grep folder1
drwxr--rwx 2 root root 4096 Dec  3 19:45 folder1
```

The following command removes existing permission for the file & set the Read permission to Others Classes.

```
chmod o=r filename/folder
```

```
root@tn:/tmp# chmod o=r folder1
root@tn:/tmp# ls -l |grep folder1
drwxr--r-- 2 root root 4096 Dec  3 19:45 folder1
```

Previously this file has read, write & execute permission for Other class. To remove Write permission & set only Read permission to Others class, we can use = symbol to do that.

Set / Remove Permission with Absolute Mode:

The following command sets the Read, Write & execute permission for User (Owner), Read & Execute permission for Group & Others Classes for the file1.txt file.

```
chmod 755 file1.txt
```

```
root@tn:/tmp# chmod 755 file1.txt
root@tn:/tmp# ls -l file1.txt
-rwxr-xr-x 1 root root 32 Dec 27 14:08 file1.txt
```

The following command sets the Read & Write permission for User (Owner), Read only permission for Group & Others Classes for the mydoc1 directory.

-R option set the same mydoc1 directory permission into all files/directories inside the mydoc1 directory.

```
chmod -R 644 mydoc1/
```

```
root@tn:/tmp# chmod -R 644 mydoc1/
root@tn:/tmp# ls -l |grep mydoc1
drw-r--r-- 2 root root 4096 Dec 27 14:11 mydoc1
root@tn:/tmp# ls -l mydoc1/
total 4
-rw-r--r-- 1 root root 331 Dec 27 14:11 file.txt
```

47. chown

The chown command is used to change the ownership & group of a file or directory.

Syntax:

```
chown [option] [Owner:Group] filename or directory name
```

Command	Explanation
chown -R	Change files and directories ownership recursively

Example:

The following example, we are changing User Ownership of **test.txt** file to **sysadmin** & Group Ownership to **IT-Team** group.

```
chown sysadmin:IT-Team test.txt
```

```
root@tn:/tmp# chown sysadmin:IT-Team test.txt
root@tn:/tmp# ls -l test.txt
-rw-r--r-- 1 sysadmin IT-Team 0 Jan  8 18:04 test.txt
```

The following example, we are changing User Ownership of **mydocs** directory and all files & directories available in mydocs directory to **trucks** & Group Ownership to **development** group.

```
chown -R trucks:development mydocs/
```

```
root@tn:/tmp# chown -R trucks:development mydocs/
root@tn:/tmp# ls -l mydocs/
total 4
-rwxr-xr-x 1 trucks development 32 Dec 27 14:08 file1.txt
root@tn:/tmp# ls -l |grep mydocs
drwxr-xr-x 2 trucks   development 4096 Jan  8 18:07 mydocs
```

6.1 Linux Access Controll Lists (ACLs):

ACLs are a second level of discretionary permissions, that may override the standard **user, group, others / read, write, execute** ones. When used correctly they can grant you a better granularity in setting access to a file or a directory, for example by giving or denying access to a specific user that is neither the file owner, nor in the group owner.

In simple terms, ACLs are used to give permission to multiple users / groups with different types of access.

There are two commands you will use when working with ACLs: **getfacl** and **setfacl**.

They are used to view and modify ACLs, respectively.

48. getfacl

The getfacl in linux used to get the file or directory access control lists(ACLs).

Syntax:

```
getfacl [option] (filename or directory name)
```

Example:

The following example, we are getting **ACL** information of the file **new.txt.**

```
getfacl new.txt
```

```
[root@redhat ~]# getfacl new.txt
# file: new.txt
# owner: root
# group: root
user::rw-
group::r--
other::r--
```

The following example, we are getting **ACL** information of the directory **trucks.**

```
getfacl trucks
```

```
[root@redhat ~]# getfacl trucks/
# file: trucks/
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
```

The following example, we are getting default **ACL** information of the directory **trucks.**

```
getfacl -d trucks
```

The following example, we are getting default **ACL** information of the directory **trucks.**

```
getfacl -R trucks/
```

Note: -R Used to get ACL information recursively. (All files available inside **trucks** directory also will list the ACL details)

```
[root@redhat ~]# getfacl -R trucks/
# file: trucks/
# owner: root
# group: root
user::rwx
group::r-x
other::r-x

# file: trucks//testacl
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
```

49. setfacl

The setfacl in linux used to set or modify the file or directory access control lists(ACLs).

Syntax:

```
setfacl [option] (filename or directory name)
```

Example:

The following file **new.txt** contains following permissions,

Owner (trucks) : Read & Write,

Group (development) : Read Only,

Others : No Permission.

```
[root@redhat ~]# ls -l new.txt
-rw-r----- 1 trucks development 0 Jan 20 10:44 new.txt
```

Set User ACL for a file:

The following example, I am granting read, write & execute acl permission to user **sysadmin** for **new.txt** file.

```
setfacl -m u:sysadmin:rwx new.txt
```

```
[root@redhat ~]# setfacl -m u:sysadmin:rwx new.txt
```

Check the ACL permission using getfacl command,

```
getfacl new.txt
```

```
[root@redhat ~]# getfacl new.txt
# file: new.txt
# owner: trucks
# group: development
user::rw-
user:sysadmin:rwx
group::r--
mask::rwx
other::---
```

Now **sysadmin** user can read, write & execute the file **new.txt**. but he is not owner or development group member.

Set Group ACL for a file:

The following example, I am granting read & execute acl permission to group **IT-Team** for **new.txt** file.

```
setfacl -m g:IT-Team:rwx new.txt
```

```
[root@redhat ~]# setfacl -m g:IT-Team:rwx new.txt
```

Note: All **IT-Team** group members will have the read & execute permission for the **new.txt** file.

Check the ACL permission using getfacl command,

```
getfacl new.txt
```

```
[root@redhat ~]# getfacl new.txt
# file: new.txt
# owner: trucks
# group: development
user::rw-
user:sysadmin:rwx
group::r--
group:IT-Team:rwx
mask::rwx
other::---
```

Now **IT-Team** group members can read, write & execute the file **new.txt** even they are not a owner or development group member.

Set Others ACL for a Directory:

The following example, Others (All users who are not owners or groups) will be given as default reading permissions for the directory called "**trucks**"

```
setfacl -d -m o:r trucks
```

-d – Set default for the files that will create in future inside **trucks** directory.

```
[root@redhat ~]# setfacl -d -m o:r trucks
```

Check the ACL permission using getfacl command,

```
getfacl trucks
```

```
[root@redhat ~]# getfacl trucks/
# file: trucks/
# owner: root
# group: root
user::rwx
group::r-x
other::r--
default:user::rwx
default:group::r-x
default:other::r--
```

Set User & Group ACL for a folder recursively:

The following example, we are going to set the following permissions directory called "**www**" Recursively.

Owners (trucks) : Read & Write

Group (development) : Read only

Others : No Permission

```
[root@redhat ~]# ls -l |grep www
drw-r-----  2 root   root           6 Jan 20 11:26 www
```

The following example, Others (All users who are not owners or groups) will be given as default reading permissions for the directory called "**www**" recursively.

```
setfacl -d -Rm o:r www
```

-R – Set the mentioned permission for the files & directories that already available inside **www** directory.

-d – Set default for the files that will create in future inside **www** directory.

```
[root@redhat ~]# setfacl -d -Rm o:r www
```

Check the ACL permission using getfacl command,

```
getfacl -R www
```

```
[root@redhat ~]# getfacl -R www
# file: www
# owner: root
# group: root
user::rw-
group::r--
other::---
default:user::rw-
default:group::r--
default:other::r--

# file: www/testacl
# owner: root
# group: root
user::rw-
group::r--
other::r--
default:user::rw-
default:group::r--
default:other::r--
```

Remove single user ACL for a file:

The following example, we are going to remove the **sysadmin** user ACL permission for the file named "**new.txt**".

```
setfacl -x u:sysadmin new.txt
```

```
[root@redhat ~]# setfacl -x u:sysadmin new.txt
```

Check the ACL permission using getfacl command,

```
getfacl new.txt
```

```
[root@redhat ~]# getfacl new.txt
# file: new.txt
# owner: trucks
# group: development
user::rw-
group::r--
group:IT-Team:rwx
mask::rwx
other::---
```

As you can see the above example, **sysadmin** user ACL permission has been removed.

Remove all ACL for a file:

The following example, we are going to remove all ACL permissions for the file named "**new.txt**".

```
setfacl -b new.txt
```

```
[root@redhat ~]# setfacl -b new.txt
```

Check the ACL permission using getfacl command,

```
getfacl new.txt
```

Remove ACL for a directory:

The following example, we are going to remove all ACL permissions & default ACL permission for the directory named "**trucks**" recursively.

```
setfacl -b -k -R trucks
```

```
[root@redhat ~]# setfacl -b -k -R trucks
```

-b – Remove all extended ACL entries.

-k – Remove the default.

-R – Remove all ACL entries from subdirectories also.

Now check the ACL permission using getfacl command,

```
getfacl trucks
```

```
[root@redhat ~]# getfacl -R trucks
# file: trucks
# owner: root
# group: root
user::rwx
group::r-x
other::r--

# file: trucks/testacl
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
```

As you can see above example, **trucks** directory & subdirectory **testacl** permission has been reset to default permissions.

--End of Chapter 6.

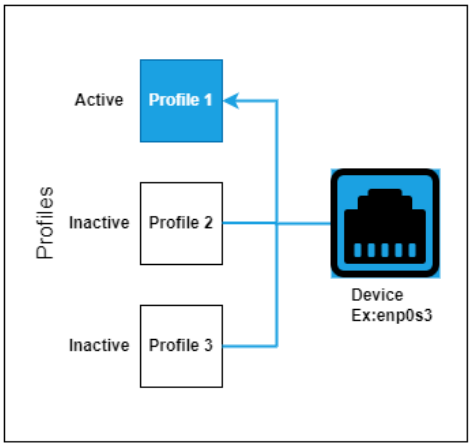
Chapter 7: Linux Network Management

7.1 Manage Network Interface in RedHat based Operating System:

Starting from Red Hat Enterprise Linux 7 onwards, the default networking service is provided by **NetworkManager.** It is a dynamic network control and configuration daemon designed to keep network devices and connections active when they are available.

Most of the server will be set to static IP address only in organizations. In this module we will see how to set static ip address in RHEL / CentOS / Fedora / Oracle / Rocky / Alma Linux OS.

Network Architecture of Network Manager:



Network Devices:

- Network devices refer to the hardware or interfaces through which a system connects to a network. Examples include Ethernet cards (eth0, eth1), Wi-Fi adapters (wlan0), and loopback devices (lo).
- NetworkManager handles the configuration and management of these devices, allowing users to control and configure network connectivity.

Network Connection Profiles:

- Network connection profiles are configurations that define how a network connection should be established. These profiles include details such as IP address settings, DNS configuration, gateway information, and more.
- NetworkManager stores these profiles, and users can create, modify, and switch between them to accommodate different network environments (e.g., home, office, public Wi-Fi).

Check The Network Manager Status:

```
systemctl status NetworkManager
```

NetworkManager offers management through different tools:

- nmtui - Network Manager Text User Interface
- nmcli - Network Manager Command Line Interface

Note that nmtui does not support all types of connections. In particular, you cannot edit VPNs, wireless network connections using WPA Enterprise, or Ethernet connections using 802.1X.

Basic prerequisites are needed to configure the network interface:

- IP Address (IPv4 / IPv6),
- Subnet,
- Gateway,
- DNS.

Configuring Network Interface using nmtui:

In this example, I'm going to configure the network interface with the following details:

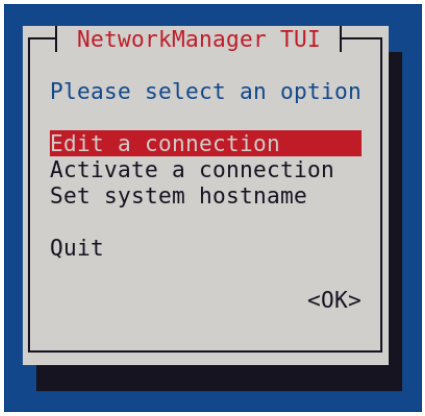
- IP: 192.168.1.100
- Subnet: 24
- Gateway: 192.168.1.1
- DNS: 8.8.8.8

Please replace the above details with your respective information.

STEP 1: Login into RHEL Based Server as a root user and in Terminal type:

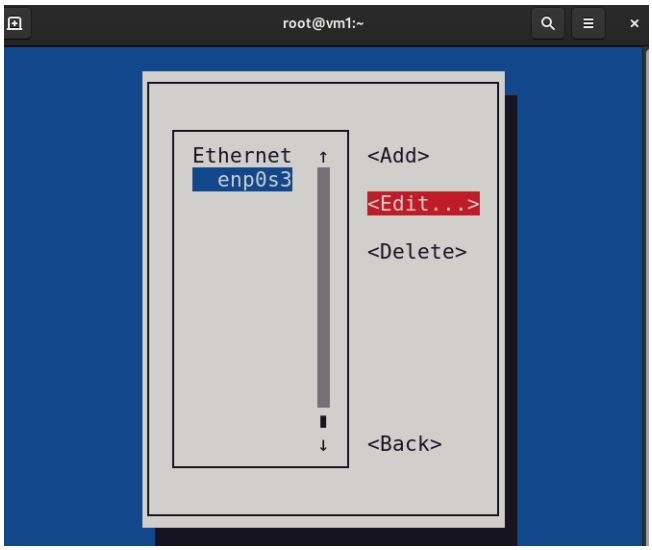
```
nmtui
```

Network manager User Interface will open.

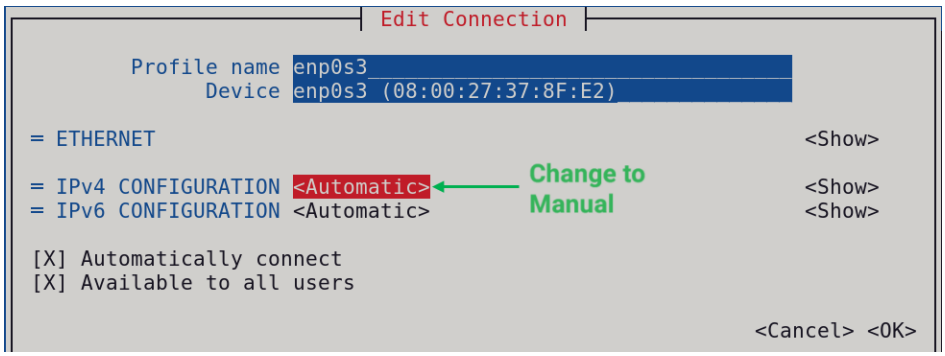


STEP 2: Select the Edit a connection menu entry and Press **Enter**.

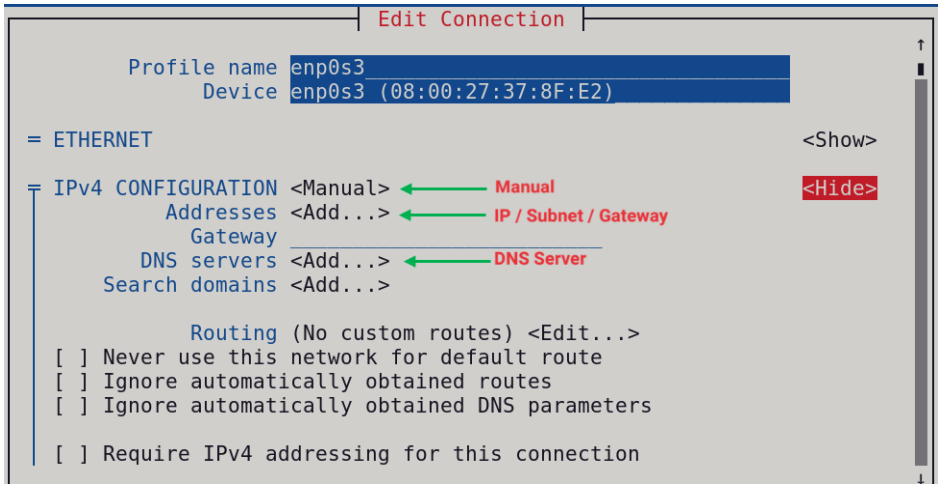
Now you can see the Ethernet adapter (E.X: enp0s3). Press the **Tab** key to navigate to the **Edit** option and press **Enter**.



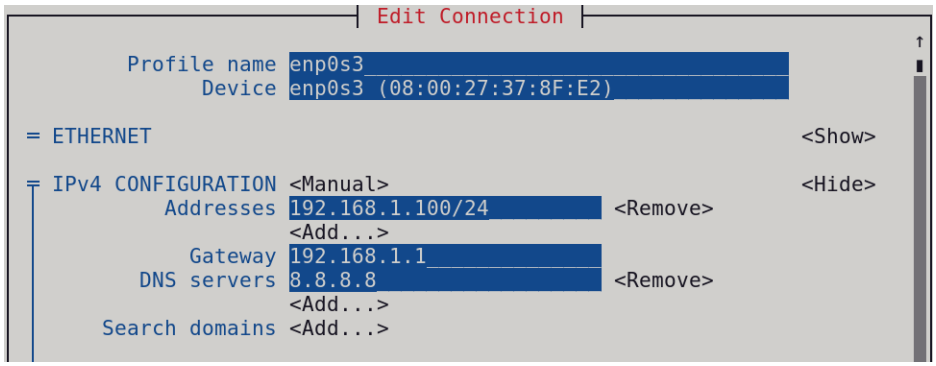
STEP 3: Change IPv4 Configuration from Automatic to Manual.



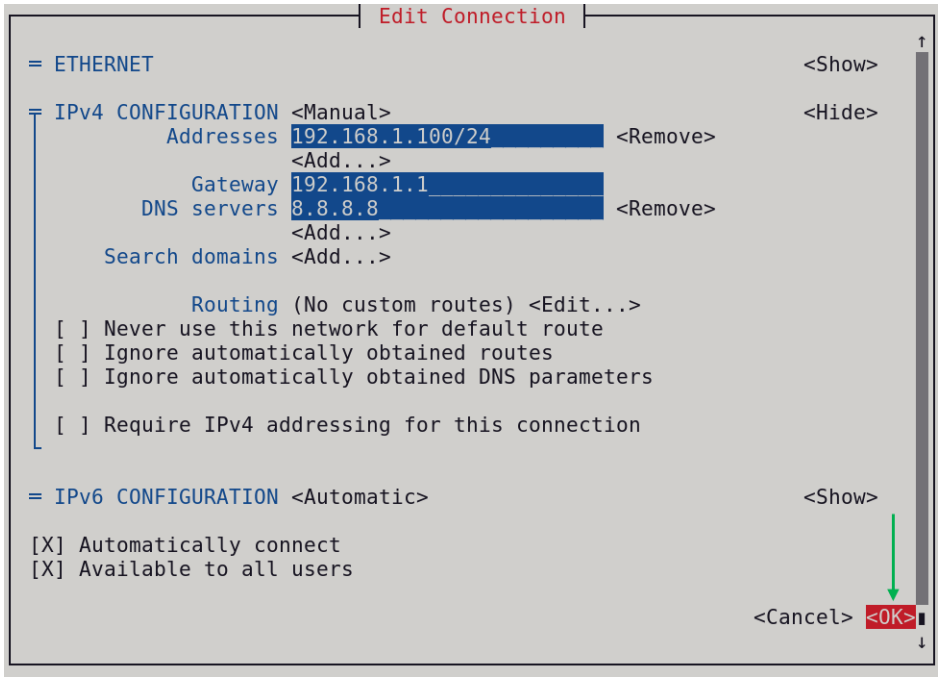
Now IP Address / Gateway / DNS Servers options will be enabled. Add your IP, Subnet, Gateway & DNS.



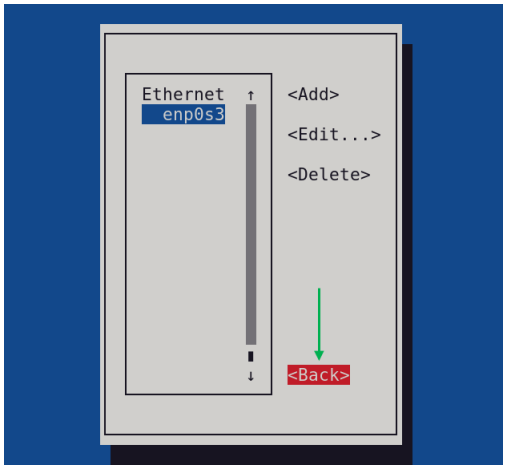
Example:



STEP 4: Scroll Down and Select **OK**.

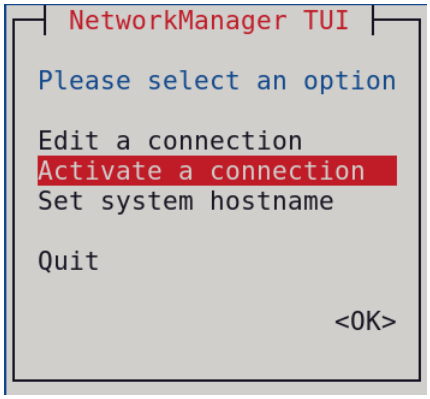


STEP 5: Now Press **Tab** Key and select **Back**.

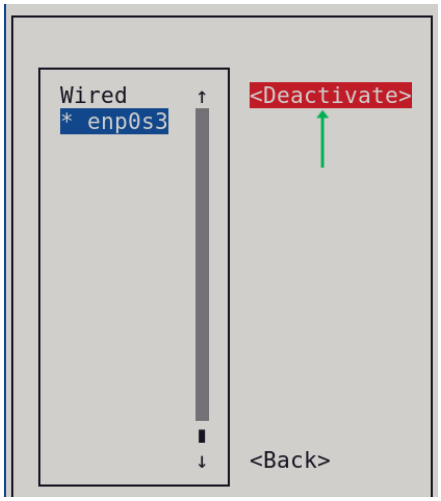


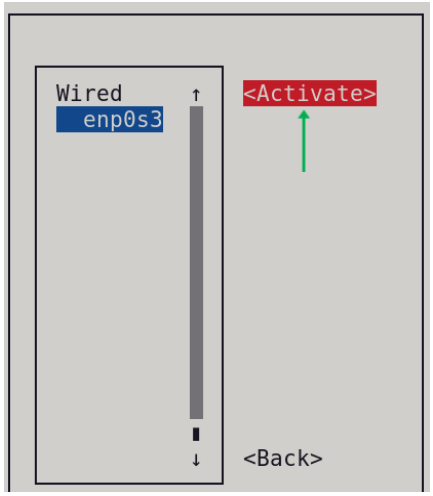
STEP 6: Now we have Reload the Network manager to changes take effect.

Select the **Activate a connection** menu entry.

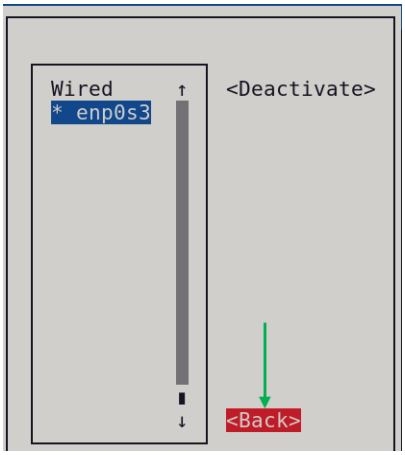


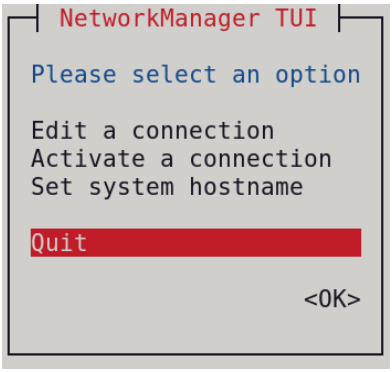
STEP 7: Select Deactivate option and **Activate** it again. This will reload the Network manager.





STEP 8: Now Select **Back** and **Quit** to Exit from the NMTUI interface.





STEP 8: Now, let's check the interface **enp0s3** to verify whether the static IP has been assigned or not with the following command. [**Note: Replace enp0s3 with your interface name]**

```
ip address show enp0s3
```

```
[root@vm1 ~]# ip address show enp0s3
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:37:8f:e2 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.100/24 brd 192.168.1.255 scope global noprefixroute enp0s3
       valid_lft forever preferred_lft forever
    inet6 2001:4490:4e95:ff61:a00:27ff:fe37:8fe2/64 scope global dynamic noprefixroute
       valid_lft 71783sec preferred_lft 71783sec
    inet6 fe80::a00:27ff:fe37:8fe2/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Great! We have successfully assigned a static IP address to our network interface using the nmtui tool.

Configuring Network Interface using nmcli:

Network Manager managing network interfaces using network profiles.

To list all the Network interfaces, run:

```
nmcli device
```

```
[root@vm1 ~]# nmcli device
DEVICE  TYPE      STATE      CONNECTION
enp0s3  ethernet  connected  enp0s3
lo      loopback  unmanaged  --
[root@vm1 ~]#
```

To list all the available profiles, run:

```
nmcli connection show
```

```
[root@vm1 ~]# nmcli connection show
NAME    UUID                                  TYPE      DEVICE
enp0s3  b19c8f8d-5baf-3973-9bbd-bf48e6573d44  ethernet  enp0s3
```

In the above snapshot, you are able to view:

- **NAME:** enp0s3 – Profile Name
- **UUID:** Unique Identification number for Profile enp0s3
- **TYPE:** ethernet – Connectivity Type is ethernet
- **DEVICE:** Ethernet Device Name is enp0s3

To view the detailed information about a profile:

```
nmcli connection show PROFILE
```

Replace PROFILE with your network profile name

Example:

```
nmcli connection show enp0s3
```

```
[root@vm1 ~]# nmcli connection show enp0s3
connection.id:                          enp0s3
connection.uuid:                        b19c8f8d-5baf-3973-9bbd-bf48e6573d44
connection.stable-id:                   --
connection.type:                        802-3-ethernet
connection.interface-name:              enp0s3
connection.autoconnect:                 yes
```

Add & Configure New Profile (Connection) Using nmcli:

In this example, I'm going to configure the network interface with the following details:

- IP: 192.168.1.200
- Subnet: 24
- Gateway: 192.168.1.1
- DNS: 8.8.8.8

Please replace the above details with your respective information.

Syntax to add a Profile:

```
nmcli connection add con-name [con_name] type [con_type] ifname [interface_name]
```

Syntax to Configure (Modify) a Profile:

```
nmcli connection mod "[Profile name]" ipv4.addresses "[IP/SUBNET]" ipv4.gateway
"[GATEWAY]" ipv4.dns "[DNS]"
```

Example:

Step 1: Create a Profile named "test_static" for the interface enp0s3.

```
nmcli connection add con-name test_static type ethernet ifname enp0s3
```

```
[root@vm1 ~]# nmcli connection add con-name test_static type ethernet ifname enp0s3
Connection 'test_static' (55eb625a-ea98-421e-9652-c43d8eadf5ff) successfully added.
[root@vm1 ~]#
```

Verify that the new profile has been added with the following command

```
[root@vm1 ~]# nmcli connection show
NAME         UUID                                  TYPE      DEVICE
enp0s3       b19c8f8d-5baf-3973-9bbd-bf48e6573d44  ethernet  enp0s3
test_static  55eb625a-ea98-421e-9652-c43d8eadf5ff  ethernet  --
[root@vm1 ~]#
```

In the above output, our new profile is not in green color, which means it's not active. We can add multiple profiles with different configurations, but only a single profile can be active as the primary connection.

Step 2: Modify the profile 'test_static" and assign the required IP, Subnet & DNS.

```
nmcli connection mod test_static ipv4.method manual ipv4.addresses 192.168.1.200/24
ipv4.gateway 192.168.1.1 ipv4.dns 8.8.8.8
```

The above command will add the IP address 192.168.1.200, subnet /24, gateway 192.168.1.1, and DNS 8.8.8.8 to the profile named 'test_static'.

```
[root@vm1 ~]# nmcli con mod test_static ipv4.method manual ipv4.addresses 192.168.1.200/24
ipv4.gateway 192.168.1.1 ipv4.dns 8.8.8.8
[root@vm1 ~]#
```

Step 3: To Activate the new profile, execute:

```
nmcli connection up test_static
```

```
[root@vm1 ~]# nmcli connection up test_static
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnectio
n/2)
[root@vm1 ~]#
```

Verify that the new profile has been activated with the following command:

```
[root@vm1 ~]# nmcli connection show
NAME         UUID                                  TYPE      DEVICE
test_static  55eb625a-ea98-421e-9652-c43d8eadf5ff  ethernet  enp0s3
enp0s3       b19c8f8d-5baf-3973-9bbd-bf48e6573d44  ethernet  --
[root@vm1 ~]#
```

Step 4: Now, let's check the interface **enp0s3** to verify whether the static IP has been assigned or not with the following command. [**Note: Replace enp0s3 with your interface name]**

```
ip address show enp0s3
```

```
[root@vm1 ~]# ip addr show enp0s3
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:37:8f:e2 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.200/24 brd 192.168.1.255 scope global noprefixroute enp0s3
       valid_lft forever preferred_lft forever
    inet6 2001:4490:4e95:8d96:3196:28b1:5983:633f/64 scope global dynamic noprefixroute
       valid_lft 47680sec preferred_lft 47680sec
    inet6 fe80::539a:4f00:7042:45de/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

How to Delete a Profile?

Syntax to Delete a Profile:

```
nmcli connection delete [Profile name]
```

Example:

Step 1: In this example, we are going to delete a Profile named “test_static” for the interface enp0s3.

```
nmcli connection delete test_static
```

```
[root@vm1 ~]# nmcli connection delete test_static
Connection 'test_static' (55eb625a-ea98-421e-9652-c43d8eadf5ff) successfully deleted.
```

Verify that the profile has been deleted with the following command:

```
nmcli connection show
```

```
[root@vm1 ~]# nmcli connection show
NAME    UUID                                  TYPE      DEVICE
enp0s3  b19c8f8d-5baf-3973-9bbd-bf48e6573d44  ethernet  enp0s3
```

7.2 How to Configure NIC in Debian based Operating System:

In this module we will see how to set static ip address in Debian based OS (Ubuntu / Linux Mint / Kali Linux).

STEP 1 : Open following file.

```
vi /etc/network/interfaces
```

STEP 2: Remove following line(If available)

```
iface ens160 inet dhcp
```

If you see the line looks like below image its configured as **dhcp.**

```
# The primary network interface
auto ens160
iface ens160 inet dhcp
```

STEP 3: Add the following lines & save the file.

```
iface ens160 inet static
        address 192.168.1.125
        netmask 255.255.255.0
        network 192.168.1.0
        broadcast 192.168.1.255
        gateway 192.168.1.1
        dns-nameservers 8.8.8.8
```

```
auto ens160
iface ens160 inet static
        address 192.168.1.125
        netmask 255.255.255.0
        network 192.168.1.0
        broadcast 192.168.1.255
        gateway 192.168.1.1
        dns-nameservers 8.8.8.8
```

STEP 4: Restart the Network service.

```
systemctl restart networking.service
```

7.3 Network Utilities:

50. ssh

ssh stands for “Secure Shell”. It is a protocol used to securely connect to a remote server/system. ssh is secure in the sense that it transfers the data in encrypted form between the host and the client.

ssh runs at TCP/IP port 22.

Syntax:

```
ssh [option] (remote IP/Host)
```

Command	Explanation
ssh host\IP	Connect to mentioned remote IP or Host in a secured shell
ssh user@host	Connect to mentioned remote user with mentioned remote IP or Host in a secured shell
ssh -p	Connect to remote machine with mentioned port (If custom port is set to remote machine)

Examples:

Note: IP & Port No can be varied as per remote machine configuration. In this example we used 127.0.0.1 as IP and 555 as Port No.

The following command used to connect remote IP 127.0.0.1 with currently logged in user. (In this example root)

```
ssh 127.0.0.1
```

```
root@tn:~# ssh 127.0.0.1
The authenticity of host '127.0.0.1 (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:W93rHFE0y9RgMZHU9uD7LpmbAzld7OzQPhQX8ehn+hc.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '127.0.0.1' (ECDSA) to the list of known hosts.
root@127.0.0.1's password:
```

The following command used to connect remote IP 127.0.0.1 with sysadmin user.

```
ssh sysadmin@127.0.0.1
```

```
root@tn:~# ssh sysadmin@127.0.0.1
sysadmin@127.0.0.1's password:
```

The following command used to connect remote IP 127.0.0.1 with sysadmin user and ssh port 2222.

```
ssh -p 2222 sysadmin@127.0.0.1
```

```
root@tn:~# ssh -p 2222 sysadmin@127.0.0.1
```

51. ping

PING (Packet INternet Groper) command is used to check the network connectivity between host and server/host.

This command takes as input the IP address or the URL and sends a data packet to the specified address with the message "PING" and get a response from the server/host this time is recorded which is called latency. Fast ping low latency means faster connection.

Ping uses ICMP(Internet Control Message Protocol) to send an ICMP echo message to the specified host if that host is available then it sends ICMP reply message.

Syntax:

```
Ping [IP] or [host name]
```

Command	Explanation
ping -c	Send number count ECHO request to mentioned address

Example:

The following command sends the data packets to **8.8.8.8** ip address until you stop the process.

```
ping 8.8.8.8
```

```
root@tn:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=55 time=63.3 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=55 time=22.4 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=55 time=30.6 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=55 time=20.1 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=55 time=21.6 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=55 time=20.6 ms
^C
--- 8.8.8.8 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5007ms
rtt min/avg/max/mdev = 20.167/29.823/63.361/15.406 ms
```

The following command sends only the 3 data packets to **8.8.8.8** ip address and stops the process.

```
ping -c 3 8.8.8.8
```

```
root@tn:~# ping -c 3 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=115 time=15.4 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=115 time=14.8 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=115 time=14.8 ms

--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 14.822/15.044/15.421/0.267 ms
```

52. traceroute

traceroute command in Linux prints the route that a packet takes to reach the host. This command is useful when you want to know about the route and about all the hops that a packet takes.

We can use this command in large networks like WAN networks, where several routers and switches are involved. It is used to trace the route to the IP packet or identify the hop where the packet is stopped.

Syntax:

```
traceroute [IP or Hostname]
```

Example:

The following command displays the network path to reach the **google.com** from linux server.

```
traceroute google.com
```

```
root@tn:~# traceroute google.com
traceroute to google.com (142.250.195.238), 30 hops max, 60 byte packets
```

53. nslookup

Nslookup (stands for “Name Server Lookup”) is a useful command for getting information from the DNS server.

It is a network administration tool for querying the Domain Name System (DNS) to obtain domain name or IP address mapping or any other specific DNS record. It is also used to troubleshoot DNS-related problems.

Syntax:

```
nslookup [IP or Hostname]
```

Example:

The following command displays the network path to reach the **google.com** from linux server.

```
nslookup google.com
```

```
root@tn:~# nslookup google.com
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.196.78
```

54. ifconfig

It reads the process information from the virtual files in /proc filesystem. /proc contains virtual files, this is the reason it's referred as a virtual file system.

ifconfig stands for interface configurator. It is one of the most basic commands used in network inspection.

Basic information displayed upon using ifconfig are:

- IP address
- MAC address
- MTU (Maximum Transmission Unit)

The ifconfig command has been deprecated and thus missing by default on some modern Linux distributions.

If ifconfig command not found error occurred, Manually we can install it with following command.

```
Installing ifconfig in Ubuntu:
apt-get install net-tools

Installing ifconfig in Redhat based OS:
yum install net-tools
```

Syntax:

```
ifconfig [options]
```

Command	Explanation
ifconfig eth0	View Network Settings of Specific Interface eth0
ifconfig eth0 up	enable a Network Interface eth0
ifconfig eth0 down	Disable a Network Interface eth0
ifconfig eth0 [IP]	Assign an IP Address to Network Interface eth0
ifconfig eth0 netmask [IP]	Assign a Netmask to Network Interface eth0
ifconfig eth0 broadcast [IP]	Assign a Broadcast to Network Interface eth0
ifconfig eth0 mtu 1000	Change MTU for a Network Interface eth0

Example:

The following command displays the interface **eth1** network settings.

```
ifconfig eth1
```

```
root@tn:~# ifconfig eth1
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.3.15  netmask 255.255.255.0  broadcast 10.0.3.255
        inet6 fe80::a00:27ff:fe55:2b76  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:55:2b:76  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 114  bytes 9460 (9.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

The following command enables the interface **eth1**.

```
ifconfig eth1 up
```

```
root@tn:~# ifconfig eth1 up
root@tn:~# ifconfig eth1
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.3.15  netmask 255.255.255.0  broadcast 10.0.3.255
        inet6 fe80::a00:27ff:fe55:2b76  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:55:2b:76  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 174  bytes 14263 (14.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

The following command disables the interface **eth1**.

```
ifconfig eth1 down
```

```
root@tn:~# ifconfig eth1 down
root@tn:~# ifconfig eth1
eth1: flags=4098<BROADCAST,MULTICAST>  mtu 1500
        inet 10.0.3.15  netmask 255.255.255.0  broadcast 10.0.3.255
        ether 08:00:27:55:2b:76  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 153  bytes 11874 (11.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

The following command set the IP address, subnet mask & broadcast IP of the interface **eth1**.

```
ifconfig eth1 10.0.3.20 netmask 255.255.255.0 broadcast 10.0.3.1
```

```
root@tn:~# ifconfig eth1 10.0.3.20 netmask 255.255.255.0 broadcast 10.0.3.1
root@tn:~# ifconfig eth1
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.3.20  netmask 255.255.255.0  broadcast 10.0.3.1
        inet6 fe80::a00:27ff:fe55:2b76  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:55:2b:76  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 216  bytes 19615 (19.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

The following command sets MTU to 1000 for the interface **eth1**.

```
ifconfig eth1 mtu 1000
```

```
root@tn:~# ifconfig eth1
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.3.20  netmask 255.255.255.0  broadcast 10.0.3.1
        inet6 fe80::a00:27ff:fe55:2b76  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:55:2b:76  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 216  bytes 19615 (19.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@tn:~# ifconfig eth1 mtu 1000
root@tn:~# ifconfig eth1
eth1: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1000
        inet 10.0.3.20  netmask 255.255.255.0  broadcast 10.0.3.1
        ether 08:00:27:55:2b:76  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 223  bytes 20227 (20.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

55. netstat

The network statistics (netstat) command is a networking tool used for troubleshooting and configuration, that can also serve as a monitoring tool for connections over the network.

Both incoming and outgoing connections, routing tables, port listening, and usage statistics are common uses for this command.

Syntax:

```
netstat [options]
```

Command	Explanation
netstat -t	Display all established tcp ports & sessions
netstat -u	Display all established udp ports & sessions
netstat -l	Display all listening ports
netstat -p	Display PID/Program name for sockets (sessions)
netstat -n	Don't resolve host name of the IP for the sockets (sessions)
netstat -r	Display routing table
netstat -i	Display interface table

Example:

The following command displays the,

- **tcp & udp** connected sessions,
- **tcp & udp** listening ports,
- **tcp & udp** Process name or ID of connected ports,
- **n-** shows the IP address instead of DNS name.

```
netstat -tulpn
```

```
root@tn:~# netstat -tulpn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:46597           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:49895           0.0.0.0:*               LISTEN      82756/rpc.statd
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      82408/rpcbind
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      43424/sshd
tcp6       0      0 :::34957                :::*                    LISTEN      -
tcp6       0      0 :::111                  :::*                    LISTEN      82408/rpcbind
tcp6       0      0 :::22                   :::*                    LISTEN      43424/sshd
tcp6       0      0 :::60632                :::*                    LISTEN      82756/rpc.statd
udp        0      0 127.0.0.1:676           0.0.0.0:*                           82756/rpc.statd
udp        0      0 0.0.0.0:752             0.0.0.0:*                           82408/rpcbind
udp        0      0 0.0.0.0:50627           0.0.0.0:*                           -
udp        0      0 0.0.0.0:57452           0.0.0.0:*                           82756/rpc.statd
udp        0      0 0.0.0.0:111             0.0.0.0:*                           82408/rpcbind
udp6       0      0 :::752                  :::*                                82408/rpcbind
udp6       0      0 :::54895                :::*                                -
udp6       0      0 :::39169                :::*                                82756/rpc.statd
udp6       0      0 :::111                  :::*                                82408/rpcbind
```

The output can be filtered with grep command, For example to check 22 port is up & running

```
netstat -tulpn |grep 22
```

```
root@tn:~# netstat -tulpn |grep 22
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      43424/sshd
tcp6       0      0 :::22                   :::*                    LISTEN      43424/sshd
root@tn:~#
```

The following command displays the routing table.

```
netstat -r
```

```
root@tn:~# netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         10.10.0.1       0.0.0.0         UG        0 0          0 ens160
localnet        *               255.255.255.0   U         0 0          0 ens160
```

The following command displays the interface table.

```
netstat -i
```

```
root@tn:~# netstat -i
Kernel Interface table
Iface   MTU Met   RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
ens160     1500 0      5930      0   1690 0          317      0      0      0 BMRU
lo        65536 0       288      0      0 0          288      0      0      0 LRU
```

56. arp

Address resolution protocol, aka ARP, is a communication protocol used in IPv4 networks. The arp protocol translates a machine's IP address into its physical address or Media Access Control (MAC) address. ARP functions with a cache or table that can be manipulated by the user to add or remove addresses.

Syntax:

```
arp [Options]
```

Command	Explanation
arp -i	Displays all arp entries for a particular interface
arp -a [IP]	Displays all arp entries for a particular address
arp -s	sets an entry in arp cache, ou need to specify the IP, MAC and interface.
arp -d	Removes an entry from arp cache

Example:

Note: Replace the example IP & MAC Address with your IP & MAC address.

To see all arp entries for a particular interface, you would use the following:

```
arp -i eth0
```

```
root@tn:~# arp -i eth0
Address                  HWtype  HWaddress           Flags Mask            Iface
_gateway                 ether   52:54:00:12:35:02   C                     eth0
```

To see arp entry for a particular ip address, you would use the following command:

```
arp -a 192.168.0.1
```

```
root@tn:~# arp -a 192.168.0.1
? (192.168.0.1) at 51:53:00:17:34:09 [ether] PERM on eth0
```

To remove an entry from the arp cache, simply use the -d flag, followed by the IP address you wish to remove. Seen here:

```
arp -d 192.168.0.1
```

```
root@tn:~# arp -d 192.168.0.1
```

Note: If you cannot delete arp entry, down the interface then delete the arp entry.

57. ip

This is the latest and updated version of ifconfig command.

IP stands for Internet Protocol. This command is used to show or configure network interface settings.

It is similar to ifconfig command but it is much more powerful with more functions and facilities attached to it.

The ip command replaces ifconfig, arp & route commands (mentioned commands functions can be done through ip command itself).

Syntax:

```
ip [options]
```

Example:

The following command to list and show all ip address associated on on all network interfaces.

```
ip a
```

```
root@tn:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b7:e7:66 brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
       valid_lft 86104sec preferred_lft 86104sec
    inet6 fe80::a00:27ff:feb7:e766/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:55:2b:76 brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet 10.0.3.15/24 brd 10.0.3.255 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe55:2b76/64 scope link
       valid_lft forever preferred_lft forever
```

The following command show the interface eth0 configuration.

```
ip a show eth0
```

```
root@tn:~# ip a show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b7:e7:66 brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
       valid_lft 86060sec preferred_lft 86060sec
    inet6 fe80::a00:27ff:feb7:e766/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

The following command show only the running interfaces

```
ip link ls up
```

```
root@tn:~# ip link ls up
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:b7:e7:66 brd ff:ff:ff:ff:ff:ff
    altname enp0s3
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:55:2b:76 brd ff:ff:ff:ff:ff:ff
    altname enp0s8
```

Assigns the IP address to the interface

The following command adds the IP address & subnet mask to the interface eth1.

```
ip a add [ip_addr/mask] dev [interface]

Example:
```

```
ip a add 192.168.1.200/24 dev eth1
```

```
root@tn:~# ip a add 192.168.1.200/24 dev eth1
root@tn:~# ip a show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:55:2b:76 brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet 192.168.1.200/24 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe55:2b76/64 scope link
       valid_lft forever preferred_lft forever
```

The following command removes / deletes the IP address from the interface eth0.

```
ip a del [ADDDRESS-HERE] dev [interface]

Example:
ip a del 192.168.1.200/24 dev eth0
```

```
root@tn:~# ip a del 192.168.1.200/24 dev eth1
root@tn:~# ip a show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:55:2b:76 brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet6 fe80::a00:27ff:fe55:2b76/64 scope link
       valid_lft forever preferred_lft forever
```

The following command used to flush \ reset the ethernet port **enp0s8**.

```
ip addr flush dev enp0s8
```

```
root@tn:~# ip addr flush dev enp0s8
root@tn:~#
```

The following command used to up or down the interface.

```
ip link set dev [interface] [up or down]

Example:
ip link set eth1 up
ip link set eth1 down
```

Up eth1:

```
ip link set eth1 up
```

Down eth1:

```
ip link set eth1 down
```

How do I change the MTU of the device?

The following command changes the eth1 mtu to 2000.

```
ip link set mtu {NUMBER} dev [DEVICE]

Example:
ip link set mtu 2000 dev eth1
```

```
root@tn:~# ip link set mtu 2000 dev eth1
root@tn:~# ifconfig eth1
eth1: flags=4099<UP,BROADCAST,MULTICAST>  mtu 2000
        inet 10.0.3.15  netmask 255.255.255.0  broadcast 10.0.3.255
        inet6 fe80::a00:27ff:fe55:2b76  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:55:2b:76  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 234  bytes 16780 (16.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

How to Manage ARP Entry with IP Command?

The following command displays the neighbour/arp cache.

```
ip neigh show
```

```
root@tn:~# ip neigh show
10.0.3.1 dev eth1  INCOMPLETE
10.0.2.2 dev eth0 lladdr 52:54:00:12:35:02 DELAY
```

The following command adds the arp entry.

```
ip neigh add [IP] lladdr [MAC] dev [DEVICE] nud {STATE}

Example:
ip neigh add 192.168.1.5 lladdr 00:1a:30:38:a8:00 dev eth1 nud perm
```

```
root@tn:~# ip neigh add 192.168.1.5 lladdr 00:1a:30:38:a8:00 dev eth1
root@tn:~# ip neigh show
192.168.1.5 dev eth1 lladdr 00:1a:30:38:a8:00 PERMANENT
10.0.3.1 dev eth1  INCOMPLETE
10.0.2.2 dev eth0 lladdr 52:54:00:12:35:02 REACHABLE
root@tn:~#
```

The following command deletes the arp entry.

```
ip neigh del {IPAddress} dev {DEVICE}

Example:
ip neigh del 192.168.1.5 dev eth0
```

```
root@tn:~# ip neigh del 192.168.1.5 dev eth1
root@tn:~# ip neigh show
10.0.3.1 dev eth1  INCOMPLETE
10.0.2.2 dev eth0 lladdr 52:54:00:12:35:02 REACHABLE
```

The following command clear the arp cache entry.

```
ip -s -s neigh flush all
```

How to change MAC address on Linux

The MAC address of a Linux network interface card (NIC) can be changed as follows:

```
##First down NIC Port##
ip link show eth1
ip link set dev eth1 down

## set new MAC address ##
ip link set dev eth1 address XX:YY:ZZ:AA:BB:CC

## UP NIC Port ##
ip link set dev eth1 up
```

```
root@tn:~# ip a show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 2000 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:55:2b:76 brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet 10.0.3.15/24 brd 10.0.3.255 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe55:2b76/64 scope link
       valid_lft forever preferred_lft forever
root@tn:~# ip link set dev eth1 down
root@tn:~# ip link set dev eth1 address 00:1a:30:38:a8:00
root@tn:~# ip link set dev eth1 up
```

Now check the mac address changed with ip a show command,

```
root@tn:~# ip a show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 2000 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:1a:30:38:a8:00 brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet 10.0.3.15/24 brd 10.0.3.255 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::21a:30ff:fe38:a800/64 scope link
       valid_lft forever preferred_lft forever
```

7.4 Network Packet Capturing Tools:

58. tcpdump

The **tcpdump** command is a most powerful and widely used command-line packets sniffer or package analyzer tool which is used to capture or filter **TCP/IP** packets that are received or transferred over a network on a specific interface.

How to Install tcpdump in Linux

Install On Debian, Ubuntu and Linux Mint

```
sudo apt-get install tcpdump
```

Install On RHEL/CentOS/Oracle/Fedora and Rocky Linux/AlmaLinux

```
sudo yum install tcpdump
```

Syntax:

```
tcpdump [Options]
```

Command	Explanation
tcpdump -i	Capture packets from the mentioned interface
tcpdump -c	Capture a specified number of packets
tcpdump -w	Save packet capturing in .pcap format
tcpdump -r	Read .pacap format saved files

Example:

The following command capture the packets only from the interface "**eth0".**

```
tcpdump -i eth0
```

```
root@tn:~# tcpdump -i eth0
```

The following command capture 5 packets only from the interface "**eth0".**

```
tcpdump -c 5 -i eth0
```

```
root@tn:~# tcpdump -c 5 -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
21:57:07.182467 IP tn.ssh > _gateway.55634: Flags [P.], seq 422997105:422997233, ack 553010294, win 63896, length 128
21:57:07.182510 IP tn.ssh > _gateway.55634: Flags [P.], seq 128:320, ack 1, win 63896, length 192
21:57:07.182680 IP tn.ssh > _gateway.55634: Flags [P.], seq 320:384, ack 1, win 63896, length 64
21:57:07.182990 IP _gateway.55634 > tn.ssh: Flags [.], ack 128, win 65535, length 0
21:57:07.182998 IP _gateway.55634 > tn.ssh: Flags [.], ack 320, win 65535, length 0
5 packets captured
30 packets received by filter
0 packets dropped by kernel
```

The following command capture the packets from the interface "**eth0"** and save the output to the capture1.pcap file**.**

```
tcpdump -w capture1.pcap -i eth0
```

```
root@tn:~# tcpdump -w capture1.pcap -i eth0
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C6 packets captured
8 packets received by filter
0 packets dropped by kernel
root@tn:~# ls -l capture1.pcap
-rw-r--r-- 1 tcpdump tcpdump 718 Jan  8 21:58 capture1.pcap
```

The following command used to read the **capture1.pcap** file**.**

```
tcpdump -r capture1.pcap
```

```
root@tn:~# tcpdump -r capture1.pcap
reading from file capture1.pcap, link-type EN10MB (Ethernet)
21:58:20.147195 IP tn.ssh > _gateway.55634: Flags [P.], seq 423001009:423001073, ack 553011478, win 63896, length 64
21:58:20.147270 IP tn.ssh > _gateway.55634: Flags [P.], seq 64:192, ack 1, win 63896, length 128
21:58:20.147512 IP tn.ssh > _gateway.55634: Flags [P.], seq 192:256, ack 1, win 63896, length 64
21:58:20.147865 IP _gateway.55634 > tn.ssh: Flags [.], ack 64, win 65535, length 0
21:58:20.147885 IP _gateway.55634 > tn.ssh: Flags [.], ack 192, win 65535, length 0
21:58:20.147888 IP _gateway.55634 > tn.ssh: Flags [.], ack 256, win 65535, length 0
```

The following command capture the packets transferred through specific port(22) from the interface "**eth0".**

```
tcpdump -i eth0 port 22 -c 5
```

```
root@tn:~# tcpdump -i eth0 port 22 -c 5
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
22:00:16.084174 IP tn.ssh > _gateway.55634: Flags [P.], seq 425729249:425729377, ack 553023766, win 63896, length 128
22:00:16.084225 IP tn.ssh > _gateway.55634: Flags [P.], seq 128:320, ack 1, win 63896, length 192
22:00:16.084384 IP tn.ssh > _gateway.55634: Flags [P.], seq 320:384, ack 1, win 63896, length 64
22:00:16.084587 IP _gateway.55634 > tn.ssh: Flags [.], ack 128, win 65535, length 0
22:00:16.084600 IP _gateway.55634 > tn.ssh: Flags [.], ack 320, win 65535, length 0
5 packets captured
23 packets received by filter
0 packets dropped by kernel
```

Note: I have added -c 5 option for capturing only 5 packets.

The following command capture the packets from the interface "**eth0"** source ip **192.168.1.30.**

```
tcpdump -i eth0 src 192.168.1.30
```

```
root@tn:~# tcpdump -i eth0 src 192.168.1.30
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
```

The following command capture the packets from the interface "**eth0"** to the IP Address **8.8.8.8.**

```
tcpdump -i eth0 dst 8.8.8.8
```

```
root@tn:~# tcpdump -i eth0 dst 8.8.8.8
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
22:05:44.001384 IP tn > dns.google: ICMP echo request, id 4, seq 1, length 64
22:05:45.003961 IP tn > dns.google: ICMP echo request, id 4, seq 2, length 64
22:05:46.005882 IP tn > dns.google: ICMP echo request, id 4, seq 3, length 64
```

--End of Chapter 7.

Chapter 8: Linux Package Management

8.1 Redhat Based Package Management:

59. yum

YUM stands for **Yellowdog Updater Modified**.

YUM is the primary package management tool for installing, updating, removing, and managing software packages in Red Hat Enterprise Linux.

YUM performs dependency resolution when installing, updating, and removing software packages.

YUM can manage packages from installed repositories in the system or from .rpm packages. The main configuration file for YUM is at /etc/yum.conf, and all the repos are at /etc/yum.repos.d.

Syntax:

```
yum [Options]
```

Command	Explanation
yum install	Install package
yum remove	Remove a package
yum update	Update one or all packages on your system
yum repolist	Display enabled software repositories
yum --enablerepo	Enable mentioned repo for a single command
yum --disablerepo	Disable mentioned repo for a single command
yum grouplist	List names of installed and available package groups
yum groupinstall	Install all packages in the selected group
yum clean all	Delete packages saved in yum cache
yum history	Display history for yum command

Example:

The following command installs the software package called **“vim”**.

```
yum install vim
```

```
[root@redhat ~]# yum install vim
Loaded plugins: fastestmirror, product-id, search-disabled-repos
Loading mirror speeds from cached hostfile
 * base: mirrors.nxtgen.com
 * epel: mirror.dimensi.cloud
 * extras: mirrors.nxtgen.com
 * updates: mirrors.nxtgen.com
Resolving Dependencies
--> Running transaction check
---> Package vim-enhanced.x86_64 2:7.4.629-8.el7_9 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================================================
 Package                        Arch                       Version                                Repository                Size
================================================================================================================================
Installing:
 vim-enhanced                   x86_64                     2:7.4.629-8.el7_9                      updates                  1.1 M

Transaction Summary
================================================================================================================================
Install  1 Package

Total download size: 1.1 M
Installed size: 2.2 M
Is this ok [y/d/N]: y
Downloading packages:
vim-enhanced-7.4.629-8.el7_9.x86_64.rpm                                                          | 1.1 MB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : 2:vim-enhanced-7.4.629-8.el7_9.x86_64                                                                        1/1
  Verifying  : 2:vim-enhanced-7.4.629-8.el7_9.x86_64                                                                        1/1

Installed:
  vim-enhanced.x86_64 2:7.4.629-8.el7_9

Complete!
```

The following command removes the software package called **"vim"**.

```
yum remove vim
```

```
[root@redhat ~]# yum remove vim
Loaded plugins: fastestmirror, product-id, search-disabled-repos
Resolving Dependencies
--> Running transaction check
---> Package vim-enhanced.x86_64 2:7.4.629-8.el7_9 will be erased
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================================================
 Package                        Arch                       Version                                Repository                Size
================================================================================================================================
Removing:
 vim-enhanced                   x86_64                     2:7.4.629-8.el7_9                      @updates                 2.2 M

Transaction Summary
================================================================================================================================
Remove  1 Package

Installed size: 2.2 M
Is this ok [y/N]: y
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Erasing    : 2:vim-enhanced-7.4.629-8.el7_9.x86_64                                                                        1/1
  Verifying  : 2:vim-enhanced-7.4.629-8.el7_9.x86_64                                                                        1/1

Removed:
  vim-enhanced.x86_64 2:7.4.629-8.el7_9

Complete!
```

The following command update the software package **"openssh"** to the latest package version if update available.

```
yum update openssh
```

```
[root@redhat ~]# yum update openssh
Loaded plugins: fastestmirror, product-id, search-disabled-repos
Loading mirror speeds from cached hostfile
 * base: mirrors.nxtgen.com
 * epel: mirror.dimensi.cloud
 * extras: mirrors.nxtgen.com
 * updates: mirrors.nxtgen.com
```

The following command will update & install all the latest patches and security updates to your system.

```
yum update
```

The following command search & displays the packages that contained the word vsftp.

```
yum search vsftp
```

```
[root@redhat ~]# yum search vsftp
Loaded plugins: fastestmirror, product-id, search-disabled-repos
Loading mirror speeds from cached hostfile
 * base: mirrors.nxtgen.com
 * epel: mirror.dimensi.cloud
 * extras: mirrors.nxtgen.com
 * updates: mirrors.nxtgen.com
====================================================================
vsftpd-sysvinit.x86_64 : SysV initscript for vsftpd daemon
vsftpd.x86_64 : Very Secure Ftp Daemon
```

The following command will list all the enabled Yum repositories in your system.

```
yum repolist
```

```
[root@redhat ~]# yum repolist
Loaded plugins: fastestmirror, product-id, search-disabled-repos
Loading mirror speeds from cached hostfile
 * base: mirrors.nxtgen.com
 * epel: download.nus.edu.sg
 * extras: mirrors.nxtgen.com
 * updates: mirrors.nxtgen.com
repo id                          repo name                                              status
base/7/x86_64                    CentOS-7 - Base                                        10,072
docker-ce-stable/7/x86_64        Docker CE Stable - x86_64                                 193
epel/x86_64                      Extra Packages for Enterprise Linux 7 - x86_64         13,731
extras/7/x86_64                  CentOS-7 - Extras                                         515
jenkins                          Jenkins-stable                                            141
pgdg-common/7/x86_64             PostgreSQL common RPMs for RHEL / CentOS 7 - x86_64       384
pgdg10/7/x86_64                  PostgreSQL 10 for RHEL / CentOS 7 - x86_64              1,153
pgdg11/7/x86_64                  PostgreSQL 11 for RHEL / CentOS 7 - x86_64              1,407
pgdg12/7/x86_64                  PostgreSQL 12 for RHEL / CentOS 7 - x86_64              1,020
pgdg13/7/x86_64                  PostgreSQL 13 for RHEL / CentOS 7 - x86_64                769
pgdg14/7/x86_64                  PostgreSQL 14 for RHEL / CentOS 7 - x86_64                498
pgdg15/7/x86_64                  PostgreSQL 15 for RHEL / CentOS 7 - x86_64                209
updates/7/x86_64                 CentOS-7 - Updates                                      4,538
repolist: 34,630
```

To install or update a particular package from a specific enabled or disabled repository, you must use --enablerepo an option in your yum command.

The following command enables the jenkins repo and updates the package.

```
yum --enablerepo=jenkins update jenkins
```

```
[root@redhat ~]# yum --enablerepo=jenkins update jenkins
Loaded plugins: fastestmirror, product-id, search-disabled-repos
Loading mirror speeds from cached hostfile
 * base: mirrors.nxtgen.com
 * epel: download.nus.edu.sg
 * extras: mirrors.nxtgen.com
 * updates: mirrors.nxtgen.com
Resolving Dependencies
--> Running transaction check
---> Package jenkins.noarch 0:2.361.4-1.1 will be updated
---> Package jenkins.noarch 0:2.375.1-1.1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package                Arch              Version              Repository      Size
================================================================================
Updating:
 jenkins                noarch            2.375.1-1.1          jenkins         89 M

Transaction Summary
================================================================================
Upgrade  1 Package
```

To install or update a particular package without a specific enabled repository, you must use --disablerepo an option in your yum command.

The following command disables the epel repo and install the package **"vstftp".**

```
yum install vsftp --disablerepo=epel
```

```
[root@redhat ~]# yum install vsftpd --disablerepo=epel
Loaded plugins: fastestmirror, product-id, search-disabled-repos
Loading mirror speeds from cached hostfile
 * base: mirrors.nxtgen.com
 * extras: mirrors.nxtgen.com
 * updates: mirrors.nxtgen.com
Resolving Dependencies
--> Running transaction check
---> Package vsftpd.x86_64 0:3.0.2-29.el7_9 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package                Arch              Version              Repository      Size
================================================================================
Installing:
 vsftpd                 x86_64            3.0.2-29.el7_9       updates        173 k

Transaction Summary
================================================================================
Install  1 Package

Total download size: 173 k
Installed size: 353 k
Is this ok [y/d/N]:
```

In Linux, a number of packages are bundled into a particular group. Instead of installing individual packages with yum, you can install a particular group that will install all the related packages that belong to the group.

The following command lists the all available group package names.

```
yum grouplist
```

```
[root@redhat ~]# yum grouplist
Loaded plugins: fastestmirror, product-id, search-disabled-repos
There is no installed groups file.
Maybe run: yum groups mark convert (see man yum)
Loading mirror speeds from cached hostfile
 * base: mirrors.nxtgen.com
 * epel: download.nus.edu.sg
 * extras: mirrors.nxtgen.com
 * updates: mirrors.nxtgen.com
Available Environment Groups:
   Minimal Install
   Compute Node
   Infrastructure Server
   File and Print Server
   Cinnamon Desktop
   MATE Desktop
   Basic Web Server
   Virtualization Host
   Server with GUI
   GNOME Desktop
   KDE Plasma Workspaces
   Development and Creative Workstation
```

The following command installs the group package **"Server with GUI"**. This group contains all the required packages for Linux GUI.

```
yum groupinstall "Server with GUI"
```

```
[root@redhat ~]# yum groupinstall "Server with GUI"
```

yum keeps all the repository enabled package data in **/var/cache/yum/** with each sub-directory by default,

The following command used to clean the yum cache.

```
yum clean all
```

```
[root@redhat ~]# yum clean all
Loaded plugins: fastestmirror, product-id, search-disabled-repos
Cleaning repos: base docker-ce-stable epel extras jenkins pgdg-common
Cleaning up list of fastest mirrors
Other repos take up 9.8 M of disk space (use --verbose for details)
```

The following command used to list all yum transactions executed in the system.

```
yum history
```

```
[root@redhat ~]# yum history
Loaded plugins: fastestmirror, product-id, search-disabled-repos
ID     | Login user               | Date and time    | Action(s)      | Altered
-------------------------------------------------------------------------------
    18 | root <root>              | 2023-01-03 17:28 | Install        |    1
    17 | root <root>              | 2023-01-03 17:27 | Erase          |    1
    16 | root <root>              | 2023-01-03 12:50 | Install        |    1
    15 | root <root>              | 2023-01-03 12:48 | Erase          |    1
    14 | root <root>              | 2023-01-03 12:08 | Install        |    4
    13 | root <root>              | 2022-11-28 12:48 | Install        |    3
    12 | root <root>              | 2022-11-28 12:41 | I, U           |    9
    11 | root <root>              | 2022-11-28 11:53 | Install        |    1
    10 | root <root>              | 2022-11-16 09:28 | Install        |    5
     9 | root <root>              | 2022-11-16 09:27 | Install        |    1
     8 | root <root>              | 2022-11-16 09:24 | Install        |   14
     7 | root <root>              | 2022-11-10 16:24 | Install        |   39
     6 | root <root>              | 2022-11-10 16:22 | Install        |    4
     5 | root <root>              | 2022-11-10 16:14 | I, O, U        |  208 EE
     4 | root <root>              | 2022-11-10 13:00 | Install        |    1
     3 | root <root>              | 2022-11-10 12:22 | Install        |   33
     2 | root <root>              | 2022-11-10 12:19 | Install        |   32
     1 | System <unset>           | 2022-11-10 12:08 | Install        |  316
history list
```

The following command used to view the information of yum transaction ID. It will show the installed / Removed / Updated packages details.

```
yum history info 10
```

```
[root@redhat ~]# yum history info 10
Loaded plugins: fastestmirror, product-id, search-disabled-repos
Transaction ID : 10
Begin time     : Wed Nov 16 09:28:13 2022
Begin rpmdb    : 443:69077f9b43ec2f0aa532eddc2e6ab0fdadc17757
End time       :            09:28:15 2022 (2 seconds)
End rpmdb      : 448:be7721f4950c4244b9080284b5264822ae5dd51f
User           : root <root>
Return-Code    : Success
Command Line   : install nginx
Transaction performed with:
    Installed     rpm-4.11.3-48.el7_9.x86_64                      @updates
    Installed     subscription-manager-1.24.51-1.el7.centos.x86_64 @updates
    Installed     yum-3.4.3-168.el7.centos.noarch                 @base
    Installed     yum-plugin-fastestmirror-1.1.31-54.el7_8.noarch  @base
Packages Altered:
    Dep-Install centos-indexhtml-7-9.el7.centos.noarch @base
    Dep-Install gperftools-libs-2.6.1-1.el7.x86_64     @base
    Install     nginx-1:1.20.1-9.el7.x86_64            @epel
    Dep-Install nginx-filesystem-1:1.20.1-9.el7.noarch @epel
    Dep-Install openssl11-libs-1:1.1.1k-4.el7.x86_64   @epel
history info
```

The following command used to undo & redo previously executed yum commands based on yum transaction ID.

```
yum history undo 10
yum history redo 10
```

```
yum history undo 10 --> Undo the yum command changes
yum history redo 10 --> Again redo the yum command changes
```

The yum history undo & redo commands are useful to track the installed packages on your system & can be easily revert back the installation or uninstallation.

60. dnf

The DNF command (Dandified yum) is the next-generation version of the traditional YUM package manager for RedHat based systems. It is the default package manager for Fedora22 & Redhat based OS's 8 above versions. It is intended to be a replacement for YUM.

Difference b/w yum & dnf

Syntax:

```
dnf [Options]
```

Command	Explanation
dnf install	Install package
dnf remove	Remove a package
dnf update	Update one or all packages on your system
dnf repolist	Display enabled software repositories
dnf grouplist	List names of installed and available package groups
dnf groupinstall	Install all packages in the selected group
dnf clean all	Delete packages saved in yum cache
dnf history	Display history for yum command

Example:

The following command installs the software package called **"vim"**.

```
dnf install vim
```

```
[root@redhat ~]# dnf install vim
Last metadata expiration check: 0:00:56 ago on Tuesday 10 January 2023 12:34:06 PM IST.
Dependencies resolved.
================================================================================================
 Package                 Arch              Version                   Repository           Size
================================================================================================
Installing:
 vim-enhanced            x86_64            2:7.4.629-8.el7_9         updates             1.1 M

Transaction Summary
================================================================================================
Install  1 Package

Total download size: 1.1 M
Installed size: 2.2 M
Is this ok [y/N]: y
Downloading Packages:
vim-enhanced-7.4.629-8.el7_9.x86_64.rpm                       2.0 MB/s | 1.1 MB     00:00
------------------------------------------------------------------------------------------------
Total                                                         1.0 MB/s | 1.1 MB     00:01
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                      1/1
  Installing       : vim-enhanced-2:7.4.629-8.el7_9.x86_64                                1/1
  Verifying        : vim-enhanced-2:7.4.629-8.el7_9.x86_64                                1/1

Installed:
  vim-enhanced-2:7.4.629-8.el7_9.x86_64

Complete!
```

The following command removes the software package called **"vim"**.

```
dnf remove vim
```

```
[root@redhat ~]# dnf remove vim
Dependencies resolved.
================================================================================================
 Package                 Arch              Version                   Repository           Size
================================================================================================
Removing:
 vim-enhanced            x86_64            2:7.4.629-8.el7_9         @System             2.2 M

Transaction Summary
================================================================================================
Remove  1 Package

Freed space: 2.2 M
Is this ok [y/N]: y
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                      1/1
  Erasing          : vim-enhanced-2:7.4.629-8.el7_9.x86_64                                1/1
  Verifying        : vim-enhanced-2:7.4.629-8.el7_9.x86_64                                1/1

Removed:
  vim-enhanced-2:7.4.629-8.el7_9.x86_64

Complete!
```

The following command update the software package **"openssh"** to the latest package version if update available.

```
dnf update openssh
```

```
[root@redhat ~]# dnf update openssh
Last metadata expiration check: 0:02:52 ago on Tuesday 10 January 2023 12:34:06 PM IST.
Dependencies resolved.
Nothing to do.
Complete!
```

Note: Above command is updating openssh package but since its already installed the latest version of package, So its showing nothing to do.

The following command will update & install all the latest patches and security updates to your system.

```
dnf update
```

```
[root@redhat ~]# dnf update
Updating Subscription Management repositories.
Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)    74% [=======================================              ] 1.6 MB/s |  11 MB     00:02 ETA
```

Note: Above command is updating all packages in linux machine, but if all packages are already update to date then it will show nothing to do.

The following command search & displays the packages that contained the word vsftp.

```
dnf search vsftp
```

```
[root@redhat ~]# dnf search vsftp
Last metadata expiration check: 0:12:26 ago on Tuesday 10 January 2023 12:34:06 PM IST.
================================================================== Name & Summary Matched: vsftp =
vsftpd-sysvinit.x86_64 : SysV initscript for vsftpd daemon
=============================================================================== Name Matched: vsftp ======
vsftpd.x86_64 : Very Secure Ftp Daemon
```

The following command will list all the enabled DNF repositories in your system.

```
dnf repolist
```

```
[root@redhat ~]# dnf repolist
Updating Subscription Management repositories.
repo id                                                     repo name
rhel-9-for-x86_64-appstream-rpms                            Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)
rhel-9-for-x86_64-baseos-rpms                               Red Hat Enterprise Linux 9 for x86_64 - BaseOS (RPMs)
[root@redhat ~]#
```

In Linux, a number of packages are bundled into a particular group. Instead of installing individual packages with dnf, you can install a particular group that will install all the related packages that belong to the group.

The following command lists the all available group package names.

```
dnf grouplist
```

```
[root@redhat ~]# dnf grouplist
Updating Subscription Management repositories.
Last metadata expiration check: 1:58:11 ago on Wed 11 Jan 2023 03:51:09 PM IST.
Available Environment Groups:
   Server with GUI
   Server
   Workstation
   Virtualization Host
   Custom Operating System
Installed Environment Groups:
   Minimal Install
Available Groups:
   Container Management
   RPM Development Tools
```

The following command installs the group package **"Server with GUI"**. This group contains all the required packages for Linux GUI.

```
dnf groupinstall "Server with GUI"
```

```
[root@redhat ~]# dnf groupinstall "Server with GUI"
Updating Subscription Management repositories.
Last metadata expiration check: 2:02:51 ago on Wed 11 Jan 2023 03:51:09 PM IST.
```

dnf keeps all the repository enabled package data in **/var/cache/dnf/** with each sub-directory by default,

The following command used to clean the dnf cache.

```
dnf clean all
```

```
[root@redhat ~]# dnf clean all
Updating Subscription Management repositories.
19 files removed
```

The following command used to list all dnf transactions executed in the system.

```
dnf history
```

```
[root@redhat ~]# dnf history
Updating Subscription Management repositories.
ID     | Command line                                                        | Date and time    | Action(s)      | Altered
---------------------------------------------------------------------------------------------------------------------------
     2 | update                                                              | 2023-01-11 17:13 | I, U           |  216 E<
     1 |                                                                     | 2022-08-30 23:48 | Install        |  360 >E
[root@redhat ~]# dnf history redo 2
Updating Subscription Management repositories.
Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)    45% [==========================                    ] 1.2 MB/s | 6.8 MB     00:06 ETA
```

The following command used to view the information of dnf transaction ID. It will show the installed / Removed / Updated packages details.

```
dnf history info 2
```

```
[root@redhat ~]# dnf history info 2
Updating Subscription Management repositories.
Transaction ID : 2
Begin time     : Wed 11 Jan 2023 05:13:54 PM IST
Begin rpmdb    : 285d44f40c3f0a26ad72cf2d855222cdf123450cfb694588b01027d8cbdbe4ec
End time       : Wed 11 Jan 2023 05:16:30 PM IST (156 seconds)
End rpmdb      : 9d1840ebb7e201feba411364bcfdc94da302dd4eb900d5359135aabf9d6c9783
User           : root <root>
Return-Code    : Success
Releasever     : 9
Command Line   : update
Comment        :
```

The following command used to undo & redo previously executed yum commands based on yum transaction ID.

```
dnf history undo 2
dnf history redo 2
```

The dnf history undo & redo commands are useful to track the installed packages on your system & can be easily revert back the installation or uninstallation.

61. rpm

RPM is a popular package management tool in Red Hat Enterprise Linux-based distros. Using RPM, you can install, uninstall, and query individual software packages.

Still, it cannot manage dependency resolution like YUM. RPM does provide you useful output, including a list of required packages.

RPM packages will be helpful to install packages on offline machines. But if the RPM Package requires an dependencies then we have to install the dependencies manually. That's the hard part.

Syntax:

```
rpm [Options]
```

Command	Explanation
rpm -i	Installs a package
rpm -U	Upgrades a package
rpm -v	Prints verbose output
rpm -h	Displays the # as a progress bar for the operation
rpm -q	Query for a package
rpm -e	Erase a package

Example:

For example if you want to install EPEL Repo or any other packages in a redhat based system. First we need download the EPEL or any other rpm package and run the following command to install the package.

i – Install a package

v – Verbose Output

h – Hash symbol for progress

```
rpm -ivh
```

```
[root@redhat ~]# rpm -ivh epel-release-latest-7.noarch.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1:epel-release-7-14                ################################# [100%]
[root@redhat ~]#
```

The following command used to Update the installed software package through rpm.

U – Update a package

v – Verbose Output

h – Hash symbol for progress

```
rpm -Uvh
```

The following command used to query about the installed rpm package.

```
rpm -qa epel-release
```

```
[root@redhat ~]# rpm -qa epel-release
epel-release-7-14.noarch
```

The following command used to erase or uninstall the rpm package epel-repo.

```
rpm -evh epel-release-7-14.norch
```

```
[root@redhat ~]# rpm -evh epel-release-7-14.noarch
Preparing...                          ################################# [100%]
Cleaning up / removing...
   1:epel-release-7-14                ################################# [100%]
[root@redhat ~]#
```

8.2 Debian Based Package Management:

62. dpkg

dpkg (Debian Package) is used to install and download the software in Debian based Linux systems.

Syntax:

```
dpkg [options] arguments
```

Command	Explanation
dpkg -i	Install a package
dpkg -r	Remove a package & dnt delete the configuration files
dpkg --purge	Remove a package & delete the configuration files
dpkg -l	list all the Debian packages
dpkg-reconfigure	Reconfigure the already installed package

Example:

if you want to install any package in a debian based system. First we need download the deb package and run the following command to install the package.

In this example, following command will install the htop software in the system.

i – Install a package

```
dpkg -i htop.deb
```

```
root@tn:~# dpkg -i htop_2.1.0-3_amd64.deb
Selecting previously unselected package htop.
(Reading database ... 98260 files and directories currently installed.)
Preparing to unpack htop_2.1.0-3_amd64.deb ...
Unpacking htop (2.1.0-3) ...
Setting up htop (2.1.0-3) ...
Processing triggers for mime-support (3.59ubuntu1) ...
Processing triggers for man-db (2.7.5-1) ...
```

If you want to uninstall \ remove the package use -r option along with dpkg.

In this example, following command will remove htop software in the system.

-r – only removes a package, it will never delete configuration files.

```
dpkg -r htop
```

```
root@tn:~# dpkg -r htop
(Reading database ... 98269 files and directories currently installed.)
Removing htop (2.1.0-3) ...
Processing triggers for man-db (2.7.5-1) ...
Processing triggers for mime-support (3.59ubuntu1) ...
```

In this example, following command will remove htop software in the system.

--purge – Removes a package and deletes all configuration files.

```
dpkg --purge htop
```

```
root@tn:~# dpkg --purge htop
(Reading database ... 98269 files and directories currently installed.)
Removing htop (2.1.0-3) ...
Processing triggers for man-db (2.7.5-1) ...
Processing triggers for mime-support (3.59ubuntu1) ...
root@tn:~#
root@tn:~# htop
-bash: /usr/bin/htop: No such file or directory
```

The following command will list all available\installed software's in the system.

```
dpkg -l
```

```
root@tn:~# dpkg -l
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name                                  Version                Architecture           Description
+++-=====================================-======================-======================-==================================================
ii  accountsservice                       0.6.40-2ubuntu11.6     amd64                  query and manipulate user account information
ii  acl                                   2.2.52-3               amd64                  Access control list utilities
ii  acpid                                 1:2.0.26-1ubuntu2      amd64                  Advanced Configuration and Power Interface event daemon
ii  adduser                               3.113+nmu3ubuntu4      all                    add and remove users and groups
ii  amd64-microcode                       3.20191021.1+really3.20 amd64                 Processor microcode firmware for AMD CPUs
```

The following command can be used to reinstall \reconfigure the already installed package.

dpkg-reconfigure reconfigures packages after they have already been installed. Pass it the names of a package or packages to reconfigure. It will ask configuration questions, much like when the package was first installed.

```
dpkg-reconfigure htop
```

```
root@tn:~# dpkg-reconfigure htop
root@tn:~#
```

63. apt & apt-get

The apt (Advanced Package Tool) is an interactive command-line tool for managing deb packages on Debian based Linux distributions.

Difference Between apt & apt-get:

Prior to Ubuntu 16.04, users regularly interacted with the APT package manager using command line tools: apt-get, apt-cache, and apt-config. Although these tools offer many functionalities, most average users did not utilize all the commands they provide.

Therefore, Linux wanted to create a simplified tool that only consisted of essential commands. With the release of Ubuntu 16.04 and Debian 8, they introduced a new command-line interface – apt.

To conclude this, apt command is designed for linux users where apt-get command used for scripts & programs used by linux in background.

The apt sources are defined in the **/etc/apt/sources.list** file and other files located in **/etc/apt/sources.list.d** directory.

Syntax:

```
apt [options]
apt-get [options]
```

Command	Explanation
apt update	Downloads info about latest versions of installed packages
apt upgrade	Download & installs the latest versions of installed packages
apt install	Install a package
apt remove	Remove a package & dnt delete the configuration files
apt purge	Remove a package & delete the configuration files
apt list	Lists all available packages

Example:

apt update command fetches the latest version information about all packages for all available sources\repositories make sure it is up to date.

```
apt update
```

```
root@tn:~# apt update
Get:1 http://security.ubuntu.com/ubuntu xenial-security InRelease [99.8 kB]
Hit:2 http://us.archive.ubuntu.com/ubuntu xenial InRelease
Get:3 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease [99.8 kB]
Get:4 https://esm.ubuntu.com/infra/ubuntu xenial-infra-security InRelease [7,524 B]
Get:5 https://esm.ubuntu.com/infra/ubuntu xenial-infra-updates InRelease [7,475 B]
Get:6 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease [97.4 kB]
Fetched 312 kB in 2s (123 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
```

apt-upgrade command install the latest version of all installed packages on ubuntu.

```
apt upgrade
apt upgrade teamviwer
```

```
root@tn:~# apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
#
# News about significant security updates, features and services will
# appear here to raise awareness and perhaps tease /r/Linux ;)
# Use 'pro config set apt_news=false' to hide this and future APT news.
```

Note: Sometime apt-upgrade command might install the latest version of kernel if upgrade available, which requires a reboot.

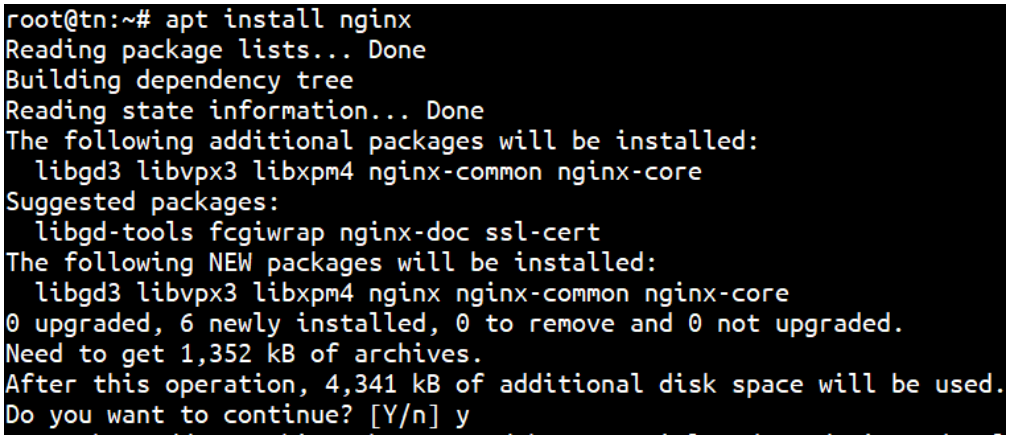
To Upgrade single package run apt upgrade followed by package name. For example. If you want to upgrade ssh package run following command.

```
apt upgrade openssh-server
```

```
root@tn:~# apt upgrade openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:7.2p2-4ubuntu2.10).
Calculating upgrade... Done
```

The apt install command used to install the new package. For example, to install nginx run following command.

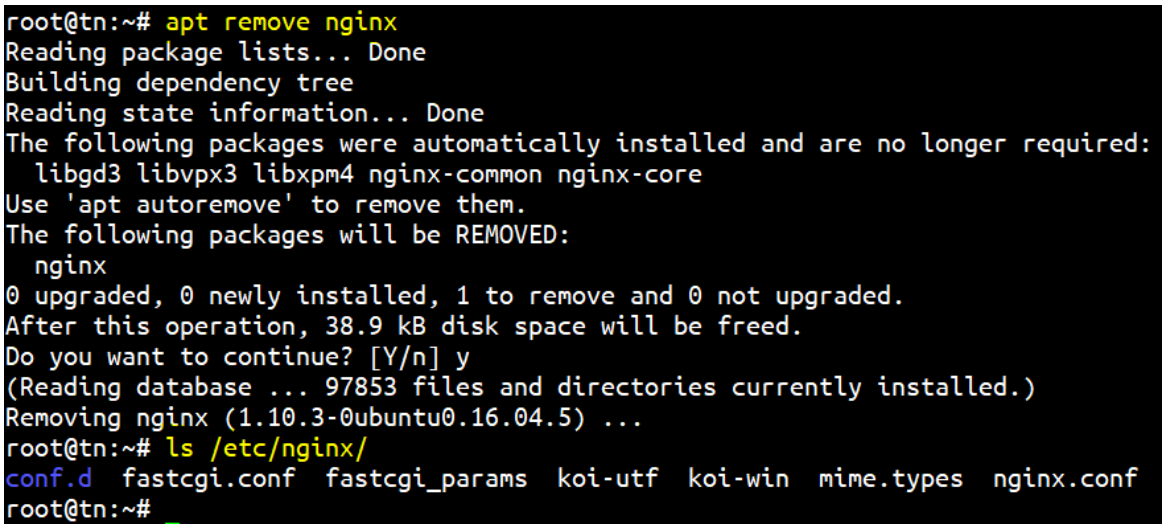
```
apt install nginx
```



apt remove command used to uninstall the already installed packages. But this will not deletes the configuration file of uninstalled package. To remove nginx package run following command.

```
apt remove nginx
```

Following example apt remove nginx command uninstalled the nginx package, but still nginx package configuration files are available in /etc/nginx path.



apt remove command used to uninstall the already installed packages. But this will deletes the configuration file of uninstalled package. To remove nginx package and delete all the config files run following command.

```
apt purge nginx*
```

```
root@tn:~# apt purge nginx*
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'nginx-full-dbg' for glob 'nginx*'
Note, selecting 'nginx-core' for glob 'nginx*'
Note, selecting 'nginx-core-dbg' for glob 'nginx*'
Note, selecting 'nginx-common' for glob 'nginx*'
Note, selecting 'nginx-doc' for glob 'nginx*'
Note, selecting 'nginx-full' for glob 'nginx*'
Note, selecting 'nginx-extras' for glob 'nginx*'
Note, selecting 'nginx-light-dbg' for glob 'nginx*'
Note, selecting 'nginx-extras-dbg' for glob 'nginx*'
Note, selecting 'nginx-light' for glob 'nginx*'
Note, selecting 'nginx' for glob 'nginx*'
The following packages were automatically installed and are no longer required:
  libgd3 libvpx3 libxpm4
Use 'apt autoremove' to remove them.
The following packages will be REMOVED:
  nginx* nginx-common* nginx-core*
0 upgraded, 0 newly installed, 3 to remove and 0 not upgraded.
After this operation, 1,485 kB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 97853 files and directories currently installed.)
Removing nginx (1.10.3-0ubuntu0.16.04.5) ...
Removing nginx-core (1.10.3-0ubuntu0.16.04.5) ...
Removing nginx-common (1.10.3-0ubuntu0.16.04.5) ...
Purging configuration files for nginx-common (1.10.3-0ubuntu0.16.04.5) ...
dpkg: warning: while removing nginx-common, directory '/var/www/html' not empty so not removed
root@tn:~# ls /etc/nginx
ls: cannot access '/etc/nginx': No such file or directory
```

The following command used to see the list of packages that can be upgraded on the system.

```
apt list --upgradable
```

```
root@tn:~# apt list --upgradable
Listing... Done
```

Note: If your system packages are upto date it will not show anything like above output.

--End of Chapter 8.

Chapter 9: Linux Firewall Management

Firewalld is an open source, host-based firewall that seeks to prevent unauthorized access to your computer.

Firewalld uses the concept of zones to segment traffic that interacts with your system. A network interface is assigned to one or more zones, and each zone contains a list of allowed ports and services.

Firewalld is the daemon's name that maintains the firewall policies. Use the firewall-cmd command to interact with the firewalld configuration.

Zone's in Firewalld:

Firewalld zones are nothing but predefined sets of rules.

Understanding predefined zones:

- block – All incoming network connections rejected. Only network connections initiated from within the system are possible.
- dmz – Classic demilitarized zone (DMZ) zone that provided limited access to your LAN and only allows selected incoming ports.
- drop – All incoming network connections dropped, and only outgoing network connections allowed.
- external – Useful for router type of connections. You need LAN and WAN interfaces too for masquerading (NAT) to work correctly.
- home – Useful for home computers such as laptops and desktops within your LAN where you trust other computers. Allows only selected TCP/IP ports.
- internal – For use on internal networks when you mostly trust the other servers or computers on the LAN.
- public – You do not trust any other computers and servers on the network. You only allow the required ports and services. For cloud servers or server hosted at your place always use public zone.
- trusted – All network connections are accepted. I do not recommend this zone for dedicated servers or VMs connected to WAN.
- work – For use at your workplace where you trust your coworkers and other servers.

You can see all zones by running the following command.

```
firewall-cmd --get-zones
```

```
[root@redhat ~]# firewall-cmd --get-zones
block dmz drop external home internal nm-shared public trusted work
```

To display the default zone, use --get-default-zone:

```
firewall-cmd --get-default-zone
```

```
[root@redhat ~]# firewall-cmd --get-default-zone
public
```

By default, firewalld is enabled and running in the public zone, all incoming traffic is rejected except SSH and DHCP.

9.1 How to enable & disable firewall service in RedHat based Operating system?

Start & Enable firewalld:

```
systemctl start firewalld
systemctl enable firewalld
```

```
[root@redhat ~]# systemctl start firewalld
[root@redhat ~]# systemctl enable firewalld
Created symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service → /usr/lib/systemd/system/firewalld.service.
Created symlink /etc/systemd/system/multi-user.target.wants/firewalld.service → /usr/lib/systemd/system/firewalld.service.
[root@redhat ~]#
```

Stop & disable firewalld:

```
systemctl stop firewalld
systemctl disable firewalld
```

```
[root@redhat ~]# systemctl stop firewalld
[root@redhat ~]# systemctl disable firewalld
Removed "/etc/systemd/system/multi-user.target.wants/firewalld.service".
Removed "/etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service".
```

9.2 How to allow / block port in RedHat based Operating system?

How to allow/open TCP/UDP port/protocol:

To open port 8080 – tcp protocol, run following command.

```
firewall-cmd --zone=public --add-port=8080/tcp --permanent
```

Note: If you didn't add –permanent option allowed port will be removed on reboot / firewalld service restart.

Reload the firewalld to configuration changes to work.

```
firewall-cmd --reload
```

To Verify the list of allowed port access, run following command.

```
firewall-cmd --list-port
```

```
[root@redhat ~]# firewall-cmd --zone=public --add-port=8080/tcp --permanent
success
[root@redhat ~]# firewall-cmd --reload
success
[root@redhat ~]# firewall-cmd --list-ports
8080/tcp
```

How to deny/block TCP/UDP port/protocol:

To deny port 8080 – tcp protocol, run following command.

```
firewall-cmd --zone=public --remove-port=8080/tcp --permanent
```

Reload the firewalld to apply the configuration changes permanently.

```
firewall-cmd --reload
```

```
[root@redhat ~]# firewall-cmd --zone=public --remove-port=8080/tcp --permanent
success
[root@redhat ~]# firewall-cmd --reload
success
[root@redhat ~]# firewall-cmd --list-ports
```

9.3 How to allow service in RedHat based Operating system?

How to allow/open http service:

To open service https, run following command.

```
firewall-cmd --zone=public --add-service=http --permanent
```

Note: If you didn't add –permanent option, allowed port will be removed on reboot / firewalld service restart.

Reload the firewalld to apply the configuration changes permanently.

```
firewall-cmd --reload
```

To Verify the list of allowed services access, run following command.

```
firewall-cmd --list-services
```

```
[root@redhat ~]# firewall-cmd --zone=public --add-service=http --permanent
success
[root@redhat ~]# firewall-cmd --reload
success
[root@redhat ~]# firewall-cmd --list-services
cockpit dhcpv6-client http ssh
```

How to deny/block http service:

To deny service https, run following command.

```
firewall-cmd --zone=public --remove-service=http --permanent
```

Reload the firewalld to configuration changes to work.

```
firewall-cmd --reload
```

We can verify the allowed services rules by following command.

```
firewall-cmd --list-services
```

```
[root@redhat ~]# firewall-cmd --zone=public --remove-service=https --permanent
success
[root@redhat ~]# firewall-cmd --reload
success
[root@redhat ~]# firewall-cmd --list-services
cockpit dhcpv6-client ssh
```

9.4 Firewall Port forwarding:

Using firewalld, you can set up ports redirection so that any incoming traffic that reaches a certain port on your system is delivered to another internal port of your choice or to an external port on another machine.

Before you redirect traffic from one port to another port, or another address, you need to know three things:

- which port the packets arrive at,
- what protocol is used,
- where you want to redirect them.

Syntax:

```
Port Forwarding to internal port:
firewall-cmd --add-forward-port=port=[port-number]:proto=[tcp|udp]:toport=[port-number]

Port Forwarding to external host & port:
firewall-cmd --add-forward-port=port=[port-number]:proto=[tcp|udp]:toport=[port-number]:toaddr=[IP]

Removing Port Forwarding:
firewall-cmd --remove-forward-port=port=[port-number]:proto=[tcp|udp]:toport=[port-number]
```

Example:

Example scenario, I have a web server(http) which is running on a port 8080. But I don't want to expose to others.

Others should access my web server with port 80, port 80 should forward this request to 8080 from a webserver.

Adding Port Forwarding:

The following command will redirect the port 8080 traffic to port 80.

```
firewall-cmd --add-forward-port=port=8080:proto=tcp:toport=80 --permanent
```

```
[root@redhat ~]# firewall-cmd --add-forward-port=port=8080:proto=tcp:toport=80 --permanent
success
[root@redhat ~]# firewall-cmd --reload
success
[root@redhat ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
        port=8080:proto=tcp:toport=80:toaddr=
  source-ports:
  icmp-blocks:
  rich rules:
```

Note: Above command will add this rule to default zone. Since we didn't mention a specific zones.

Removing Port Forwarding:

```
 firewall-cmd --remove-forward-port=port=8080:proto=tcp:toport=80 --permanent
```

We can verify the allowed port forward rules by following command.

```
firewall-cmd --list-forward-ports
or
firewall-cmd --list-all
```

```
[root@redhat ~]# firewall-cmd --remove-forward-port=port=8080:proto=tcp:toport=80 --permanent
success
[root@redhat ~]# firewall-cmd --reload
success
[root@redhat ~]# firewall-cmd --list-forward-ports

[root@redhat ~]#
```

9.5 How to enable & disable firewall service(ufw) in Debian based Operating system?

ufw – Uncomplicated Firewall.

The default firewall configuration tool for Ubuntu is ufw. Developed to ease iptables firewall configuration, ufw provides a user-friendly way to create an IPv4 or IPv6 host-based firewall.

ufw by default is initially disabled.

Enable ufw:

```
ufw enable
```

```
root@tn:~# ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
root@tn:~# ufw status
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
```

Disable ufw:

```
ufw disable
```

```
root@tn:~# ufw disable
Firewall stopped and disabled on system startup
root@tn:~# ufw status
Status: inactive
```

9.6 How to allow / deny & delete a port based rule in (ufw) Debian based Operating system?

For example, if you want to allow port 80 run the following command.

```
ufw allow 80/tcp
```

```
root@tn:~# ufw allow 80/tcp
Rule added
Rule added (v6)
root@tn:~# ufw status
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
80/tcp                     ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
80/tcp (v6)                ALLOW       Anywhere (v6)
```

Like allowing the port, if you want to deny \ block port 80 run the following command.

```
ufw deny 80/tcp
```

```
root@tn:~# ufw deny 80/tcp
Rule updated
Rule updated (v6)
root@tn:~# ufw status
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
80/tcp                     DENY        Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
80/tcp (v6)                DENY        Anywhere (v6)
```

If you want to delete a rule, for example to delete deny port 80 rule run the following command.

```
ufw delete deny 80/tcp
```

```
root@tn:~# ufw delete deny 80/tcp
Rule deleted
Rule deleted (v6)
root@tn:~# ufw status
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
```

If you want to allow specific IP Address to access the specified port, run the following command.

The following command will allow 10.10.x.x to access port 22 (SSH).

Syntax:

```
ufw allow from [source IP] to [any or destination IP] proto [tcp or udp] port [No]
```

Example:

```
ufw allow from 10.10.1.23 to any proto tcp port 22
```

```
root@tn:~# ufw allow from 10.10.1.23 to any proto tcp port 22
Rule added
root@tn:~# ufw status
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
22/tcp                     ALLOW       10.10.1.23
OpenSSH (v6)               ALLOW       Anywhere (v6)
```

9.7 How to allow / deny a app(service) in (ufw) Debian based Operating system?

In ufw we can directly add the app that opens the port. For example instead of allowing port 22 we can directly allow app ssh in ufw.

To view the list of apps, run following command. It will list all the installed apps [Ex: ssh, apache2]

```
ufw app list
```

To allow specific app, run following command.

The following example, allowing ssh app in ufw.

```
ufw allow Openssh
```

```
root@tn:~# ufw app list
Available applications:
  OpenSSH
root@tn:~# ufw allow Openssh
Rule added
Rule added (v6)
root@tn:~# ufw status
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
```

Like allowing the app, if you want to deny \ block app ssh run the following command.

```
ufw deny Openssh
```

```
root@tn:~# ufw deny Openssh
Rule updated
Rule updated (v6)
root@tn:~# ufw status
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    DENY        Anywhere
OpenSSH (v6)               DENY        Anywhere (v6)
```

Note: If you deny Openssh you wont able to take ssh session anymore until allow the Openssh. So, its not recommended to do.

If you want to allow specific IP Address to access the specified app, run the following command.

The following command will allow 10.10.x.x to access app ssh.

Syntax:

```
ufw allow from [source IP] to [any or destination IP] app [app name]
```

Example:

```
ufw allow from 10.10.1.23 to any app Openssh
```

```
root@tn:~# ufw allow from 10.10.1.23 to any app Openssh
Rule added
root@tn:~# ufw status
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
OpenSSH                    ALLOW       10.10.1.23
OpenSSH (v6)               ALLOW       Anywhere (v6)
```

Note: There is no need to specify the protocol for the application, because that information is detailed in the profile. Also, note that the app name replaces the port number.

--End of Chapter 9.

Chapter 10: Linux Disk Partition Management – RHEL Systems

Partitioning a Disk in Linux:

Disk partitioning allows system administrators to divide a hard drive into multiple logical storage units, referred as partitions.

By separating a disk into multiple partitions, system administrators can use different partitions to perform different functions.

Types of Partition tables:

There are two main types of partition table available. They are,

- Maste Boot Record (MBR)
- GUID Partition Table (GPT)

File System types:

Two types of file systems are mostly \ frequently used by linux operating systems,

XFS file system – Default file system for RHEL Based OS (RHEL, CentOS, Rocky Linux, Oracle Linux, etc).

EXT4 file system – Default file system for Debian Based OS (Ubuntu, Kali Linux, Linux Mint, etc).

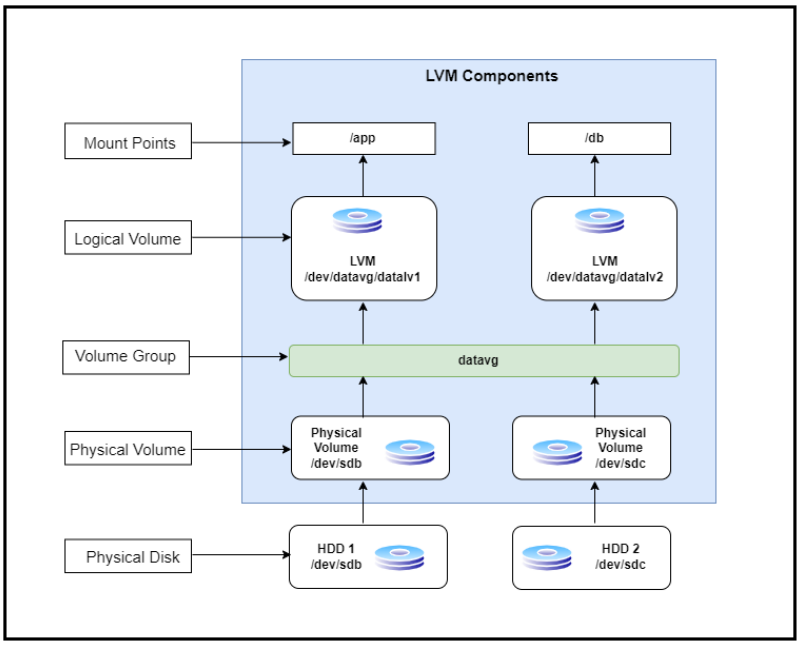
Logical Volume Manager:

LVM is a tool for logical volume management which includes allocating disks, striping, mirroring and resizing logical volumes.

- Storage volumes created under the control of the logical volume manager can be resized.
- You can think of LVM as "dynamic partitions", meaning that you can create/resize/delete LVM "partitions" (they're called "Logical Volumes" in LVM-speak) from the command line while your Linux system is running.
- No need to reboot the system to make the kernel aware of the newly created or resized partitions.

Three Components of LVM:

1. Physical Volume
2. Volume Group
3. Logical Volume



1. Physical Volume:

A physical volume is any physical storage device, such as a Hard Disk Drive (HDD), Solid State Drive (SSD), or partition, that has been initialized as a physical volume with LVM. Without properly initialized physical volumes, you cannot create Volume Groups or logical volumes.

The following syntax used to create the Physical Volume,

```
pvcreate /dev/hdd-label
```

The following syntax used to list the Physical Volume,

```
pvs (or) pvdisplay
```

2. Volume Group:

A volume group (VG) is the central unit of the Logical Volume Manager (LVM) architecture. It is what we create when we combine multiple physical volumes to create a single storage structure, equal to the storage capacity of the combined physical devices.

The following syntax used to create the Volume Group,

```
vgcreate vg-name /dev/pv-name
```

[**Note:** Replace vg-name & pv-name with appropriate values]

The following syntax used to list the Physical Volume,

```
vgs (or) vgdisplay
```

3. Logical Volume:

Logical Volumes (LV) are the final storage unit in the standard LVM architecture. These units are created from the volume group, which is made up of physical volumes (PV).

The following syntax used to create the Logical Volume,

```
Lvcreate -n lv-name -size 2G vg-name
```

[**Note:** Replace vg-name & pv-name with appropriate values]

The following syntax used to list the Logical Volume,

```
lvs (or) lvdisplay
```

10.1 How to Create a new partition in RHEL:

Step 1: Add a disk into your redhat machine and verify disk is added.

The following command list the all disks in your linux system.

```
fdisk -l
```

```
[root@redhat ~]# fdisk -l
Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x581efd55

Device     Boot  Start      End  Sectors  Size Id Type
/dev/sda1  *      2048   976895   974848  476M 83 Linux
/dev/sda2       976896 41943039 40966144 19.5G 8e Linux LVM


Disk /dev/sdb: 2 GiB, 2147483648 bytes, 4194304 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

Step 2: Create new Partition.

The following command will open **fdisk** console for **/dev/sdb** disk.

```
fdisk /dev/sdb
```

To create new partition, type

```
n
```

```
[root@redhat ~]# fdisk /dev/sdb

Welcome to fdisk (util-linux 2.37.4).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x7dc65254.

Command (m for help): n
```

Specify the type of partition using the **p** for primary and **e** for extended. This will create a primary partition.

```
Partition type: p
```

```
Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
```

The console will prompt for the number to be given to the partition. We can give any number from 1 to 4. I am giving 1 as a partition number because this is the first partition for **/dev/sdb** disk.

```
1
```

Specify the size of the partition (First sector & Last sector),

```
Example:
First sector: 2048
Last sector:
[Note: If you didn't mention Last sector size, it will allocate all free space to this partition]
```

Pressing **enter** will create our 1st partition successfully with 2GB size.

write the changes to the disk using the **w** command, else use the **q** command to quit without writing.

```
w
```

```
[root@redhat ~]# fdisk /dev/sdb

Welcome to fdisk (util-linux 2.37.4).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x3b62d46a.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-4194303, default 2048): 2048
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-4194303, default 4194303):

Created a new partition 1 of type 'Linux' and of size 2 GiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

Step 3: Create file system for new partition. Inorder to specify the file system to be used in each partition,

we can use the mkfs (make file system) command.

The following example we are creating **xfs** file system.

```
mkfs.xfs /dev/sdb1
```

```
[root@redhat ~]# mkfs.xfs /dev/sdb1
meta-data=/dev/sdb1              isize=512    agcount=4, agsize=131008 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=1 inobtcount=1
data     =                       bsize=4096   blocks=524032, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

Step 4: Mount the file system.

Create new directory.

```
mkdir /app
```

```
[root@redhat ~]# mkdir /app
```

Mount **/dev/sdb1** to **/app** directory.

```
mount /dev/sdb1 /app
```

```
[root@redhat ~]# mount /dev/sdb1 /app
```

Check the mounting.

```
df -h
```

```
[root@redhat ~]# df -h
Filesystem             Size  Used Avail Use% Mounted on
devtmpfs               4.0M     0  4.0M   0% /dev
tmpfs                  886M     0  886M   0% /dev/shm
tmpfs                  355M  5.0M  350M   2% /run
/dev/mapper/rhel-root   16G  1.3G   15G   8% /
/dev/sda1              471M  270M  202M  58% /boot
tmpfs                  178M     0  178M   0% /run/user/0
/dev/sdb1              2.0G   47M  2.0G   3% /app
```

Step 5: Mount the file system permanently.

All these mounting are temporary by default. Once we reboot the system, mounting will be reverted. To make it permanent, we must edit the File System Table of the Operating System.

```
vi /etc/fstab
```

```
[root@redhat ~]# vi /etc/fstab
```

[**Note:** A small error in this file can cause the system to be unbootable and can make the entire system to be useless. So edit carefully]

Add our mounted file systems details in the **/etc/fstab** file & save the file,

```
/dev/sdb1   /app   xfs   defaults 0 0
```

```
#
# /etc/fstab
# Created by anaconda on Tue Aug 30 18:10:19 2022
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/rhel-root   /                       xfs     defaults        0 0
UUID=aef32152-840b-4c40-b231-b9eb44f66fab /boot                   xfs     defaults        0 0
/dev/mapper/rhel-swap   none                    swap    defaults        0 0
/dev/sdb1               /app                    xfs     defaults        0 0
```

10.2 How to delete a partition in RHEL:

In this example, we are going to delete **/app** mount point.

/app is mounted with **/dev/sdb** physical disk.

Step 1: First unmount the file system.

The following command will unmount the **/app** partition in your linux system.

```
unmount /app
```

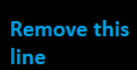
```
[root@redhat ~]# umount /app
[root@redhat ~]# df -h
Filesystem              Size  Used Avail Use% Mounted on
devtmpfs                4.0M     0  4.0M   0% /dev
tmpfs                   886M     0  886M   0% /dev/shm
tmpfs                   355M  5.0M  350M   2% /run
/dev/mapper/rhel-root    16G  1.3G   15G   8% /
/dev/sda1               471M  270M  202M  58% /boot
tmpfs                   178M     0  178M   0% /run/user/0
```

Step 2: Remove the file system entry from **/etc/fstab** file.

Delete the following line from **/etc/fstab** file.

```
/dev/sdb1   /app   xfs   defaults 0 0
```

```
#
# /etc/fstab
# Created by anaconda on Tue Aug 30 18:10:19 2022
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/rhel-root   /                       xfs     defaults        0 0
UUID=aef32152-840b-4c40-b231-b9eb44f66fab /boot                   xfs     defaults        0 0
/dev/mapper/rhel-swap   none                    swap    defaults        0 0
/dev/sdb1               /app                    xfs     defaults        0 0
~
```



Step 3: Delete the partition.

The following command will open the **fdisk** console for **/dev/sdb** disk.

```
fdisk /dev/sdb

# To delete a partition type,
d

# It will prompt the partition number you want to delete(type the number). In this
example partition no 1 only available and selected by default.
1
```

```
# Write the changes to the disk using the w command, else use the q command to quit without writing.
w
```

```
[root@redhat ~]# fdisk /dev/sdb

Welcome to fdisk (util-linux 2.37.4).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): d
Selected partition 1
Partition 1 has been deleted.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

10.3 How to Create partition using LVM in RHEL:

Step 1: Add a disk into your redhat machine and verify disk is added.

The following command list all disks in your linux system.

```
fdisk -l
```

```
[root@redhat ~]# fdisk -l
Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x581efd55

Device     Boot  Start      End  Sectors  Size Id Type
/dev/sda1  *      2048   976895   974848  476M 83 Linux
/dev/sda2       976896 41943039 40966144 19.5G 8e Linux LVM


Disk /dev/sdb: 2 GiB, 2147483648 bytes, 4194304 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

Step 2: Create Physical volume(PV) with **pvcreate** command.

```
pvcreate /dev/sdb
```

[**Note:** Replace **/dev/sdb** with appropriate partition name.]

```
[root@redhat ~]# pvcreate /dev/sdb
  Physical volume "/dev/sdb"  successfully created.
```

Verify the physical volume details using following command.

```
pvdisplay
```

```
[root@redhat ~]# pvdisplay
  "/dev/sdb"  is a new physical volume of "<2.00 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/sdb
  VG Name
  PV Size               <2.00 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               jtfBr1-r0fj-knXw-VnN9-vFaE-plAd-RR2PzW
```

Step 3: Create Volume Group (VG) with **vgcreate** command. In this example I am creating volume group named **vg-1.**

```
vgcreate vg-1 /dev/sdb
```

[**Note:** Replace **/dev/sdb** with appropriate pv name.]

```
[root@redhat ~]# vgcreate vg-1 /dev/sdb
  Volume group "vg-1" successfully created
```

Verify the Volume Group **vg-1** details using following command.

```
vgdisplay
```

```
[root@redhat ~]# vgdisplay
  --- Volume group ---
  VG Name               vg-1
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               <2.00 GiB
  PE Size               4.00 MiB
  Total PE              511
  Alloc PE / Size       0 / 0
  Free  PE / Size       511 / <2.00 GiB
  VG UUID               O8Tjzg-X3Oe-KVri-kT9F-96ec-FiFt-GSbuqb
```

Step 4: Create Logical Volume (LV) with **lvcreate** command. It will create logical volume named **lv-1** with 1.99G**.**

```
lvcreate -L 2G -n lv-1 vg-1
```

```
[root@redhat ~]# lvcreate -L 2G -n lv-1 vg-1
  Logical volume "lv-1" created.
```

Verify the Logical volume **/dev/vg-1/lv-1** details using command.

```
lvdisplay
```

```
[root@redhat ~]# lvdisplay
  --- Logical volume ---
  LV Path                /dev/vg-1/lv-1
  LV Name                lv-1
  VG Name                vg-1
  LV UUID                Fxl3Sn-yer9-0Pcp-sPQJ-IHhK-1qPV-wwWfuA
  LV Write Access        read/write
  LV Creation host, time redhat, 2023-01-18 10:46:09 +0530
  LV Status              available
  # open                 0
  LV Size                2.00 GiB
  Current LE             512
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:2
```

Step 5: Create file system for new lvm partition

The following example we are creating **xfs** file system.

```
mkfs.xfs /dev/vg-1/lv-1
```

```
[root@redhat ~]# mkfs.xfs /dev/vg-1/lv-1
meta-data=/dev/vg-1/lv-1         isize=512    agcount=4, agsize=131072 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=1 inobtcount=1
data     =                       bsize=4096   blocks=524288, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

Step 6: Mount the file system.

Create new directory.

```
mkdir /app
```

Mount **/dev/vg-1/lv-1** to **/app** directory.

```
mount /dev/vg-1/lv-1 /app
```

Check the mounting.

```
df -h
```

```
[root@redhat ~]# mount /dev/vg-1/lv-1 /app
[root@redhat ~]# df -h
Filesystem               Size  Used Avail Use% Mounted on
devtmpfs                 4.0M     0  4.0M   0% /dev
tmpfs                    886M     0  886M   0% /dev/shm
tmpfs                    355M  5.0M  350M   2% /run
/dev/mapper/rhel-root     16G  1.3G   15G   8% /
/dev/sda1                471M  270M  202M  58% /boot
tmpfs                    178M     0  178M   0% /run/user/0
/dev/mapper/vg--1-lv--1  2.0G   47M  2.0G   3% /app
```

Step 7: Mount the file system permanently.

All these mounting are temporary by default. Once we reboot the system, mounting will be reverted. To make it permanent, we must edit the File System Table of the Operating System.

```
vi /etc/fstab
```

```
[root@redhat ~]# vi /etc/fstab
```

[**Note:** A small error in this file can cause the system to be unbootable and can make the entire system to be useless. So edit carefully]

Add our mounted file systems details in the **/etc/fstab** file & save the file,

```
/dev/vg-1/lv-1   /app   xfs   defaults 0 0
```

```
# /etc/fstab
# Created by anaconda on Tue Aug 30 18:10:19 2022
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/rhel-root   /                       xfs     defaults        0 0
UUID=aef32152-840b-4c40-b231-b9eb44f66fab /boot                   xfs     defaults        0 0
/dev/mapper/rhel-swap   none                    swap    defaults        0 0
/dev/vg-1/lv-1          /app                    xfs     defaults        0 0
~
```

10.4 How to Extend partition size using LVM in RHEL:

In this example, we are going to increase the size of **/app** mount point size from 2GB to 5GB.

/app mounted with **/dev/vg-1/lv-1** (VG Name – vg-1, LV Name – lv-1).

Step 1: Add a disk into your redhat machine and verify disk is added.

The following command list the all disks in your linux system.

```
fdisk -l
```

Step 2: Create Physical volume(PV) with **pvcreate** command.

```
pvcreate /dev/sdc
```

[**Note:** Replace **/dev/sdc** with appropriate partition name.]

```
[root@redhat ~]# pvcreate /dev/sdc
  Physical volume "/dev/sdc"  successfully created.
```

Verify the physical volume details using following command.

```
pvdisplay
```

```
  "/dev/sdc"  is a new physical volume of "<3.01 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/sdc
  VG Name
  PV Size               <3.01 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               ee3Blp-FAVx-8vu2-BVEV-wcyB-cJfr-dy8Qbe
```

Step 4: Extend the volume group **vg-1**.

```
vgextend vg-1 /dev/sdc
```

[**Note:** Replace VG Name **vg-1** & PV Name **/dev/sdc** with appropriate VG & PV name.]

As you can see, Volume Group **vg-1** size has been extended to 5GB.

```
[root@redhat ~]# vgextend vg-1 /dev/sdc
  Volume group "vg-1" successfully extended
[root@redhat ~]#
[root@redhat ~]# vgs
  VG   #PV #LV #SN Attr   VSize  VFree
  rhel   1   2   0 wz--n- 19.53g     0
  vg-1   2   1   0 wz--n- <5.02g <3.02g
```

Step 5: Now Extend the logical volume.

```
lvextend -L +3G /dev/vg-1/lv-1
```

[**Note:** Replace LV Name **/dev/vg-1/lv-1** with appropriate lv name.]

```
[root@redhat ~]# lvextend -L +3G /dev/vg-1/lv-1
  Size of logical volume vg-1/lv-1 changed from 2.00 GiB (512 extents) to 5.00 GiB (1280 extents).
  Logical volume vg-1/lv-1 successfully resized.
[root@redhat ~]#
[root@redhat ~]# lvs
  LV   VG   Attr       LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  root rhel -wi-ao---- 15.80g
  swap rhel -wi-ao---- <3.73g
  lv-1 vg-1 -wi-ao----  5.00g
```

Step 6: Resize the file system.

```
xfs_growfs /dev/vg-1/lv-1
```

[**Note:** Replace LV Name **/dev/vg-1/lv-1** with appropriate lv name.]

```
[root@redhat ~]# xfs_growfs /dev/vg-1/lv-1
meta-data=/dev/mapper/vg--1-lv--1 isize=512    agcount=4, agsize=131072 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=1 inobtcount=1
data     =                       bsize=4096   blocks=524288, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
data blocks changed from 524288 to 1310720
```

Now check the **/app** mount directory size with **df** command.

```
 df -h /app
```

```
[root@redhat ~]# df -h /app
Filesystem              Size  Used Avail Use% Mounted on
/dev/mapper/vg--1-lv--1  5.0G   69M  5.0G   2% /app
```

10.5 How to reduce LVM partition size in RHEL:

Now we are going to reduce 3GB size of the mount point **/app** directory, **/app** is mounted with **/dev/vg-1/lv-1**.

Here **lv-1** is the logical volume & **vg-1** is the Volume Group.

Before reducing the lvm size, it is always good to backup the data, so that it will not be a data loss if something goes wrong.

To Reduce a logical volume there are 6 steps needed to be done very carefully.

Step 1: Take xfs file system backup. Because in xfs file system we need to take backup of filesystem otherwise all data will be wiped out.

The following command used to take **xfsdump** backup for the **/app** directory. This command will create backup for **/app** directory in the destination **/tmp/** directory. Backup file name is **app.dump.**

```
xfsdump -l 0 -L "app_dir_backup" -f /tmp/app.dump /app
```

```
[root@redhat ~]# xfsdump -l 0 -L "app_dir_backup" -f /tmp/app.dump /app
xfsdump: using file dump (drive_simple) strategy
xfsdump: version 3.1.10 (dump format 3.0) - type ^C for status and control
xfsdump: level 0 dump of redhat:/app
xfsdump: dump date: Thu Jan 19 12:22:01 2023
xfsdump: session id: 0009118b-438a-4f68-9912-d0a127e2b095
xfsdump: session label: ""app_dir_backup""
xfsdump: ino map phase 1: constructing initial dump list
xfsdump: ino map phase 2: skipping (no pruning necessary)
xfsdump: ino map phase 3: skipping (only one dump stream)
xfsdump: ino map construction complete
xfsdump: estimated dump size: 24320 bytes

 ============================ media label dialog ============================

please enter label for media in drive 0 (timeout in 300 sec)
 -> app-dir-data
media label entered: "app-dir-data"

 -------------------------------- end dialog --------------------------------

xfsdump: creating dump session media file 0 (media 0, file 0)
xfsdump: dumping ino map
xfsdump: dumping directories
xfsdump: dumping non-directory files
xfsdump: ending media file
xfsdump: media file size 24464 bytes
xfsdump: dump size (non-dir files) : 0 bytes
xfsdump: dump complete: 10 seconds elapsed
xfsdump: Dump Summary:
xfsdump:   stream 0 /tmp/app.dump OK (success)
xfsdump: Dump Status: SUCCESS
[root@redhat ~]#
```

Step 2: Unmount the mount point directory (**/app** directory).

```
umount /app
```

```
[root@redhat ~]# umount /app
[root@redhat ~]#
```

Step 3: Now reduce the logical volume size using following command. In this command we are reducing 3GB.

```
lvreduce -L -3G /dev/vg-1/lv-1
```

```
[root@redhat ~]# lvreduce -L -3G /dev/vg-1/lv-1
  WARNING: Reducing active logical volume to 2.00 GiB.
  THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce vg-1/lv-1? [y/n]: y
  Size of logical volume vg-1/lv-1 changed from 5.00 GiB (1280 extents) to 2.00 GiB (512 extents).
  Logical volume vg-1/lv-1 successfully resized.
```

Step 4: Make File System for Logical Volume using following command.

```
mkfs.xfs -f /dev/vg-1/lv-1
```

```
[root@redhat ~]# mkfs.xfs -f /dev/vg-1/lv-1
meta-data=/dev/vg-1/lv-1         isize=512    agcount=4, agsize=131072 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=1 inobtcount=1
data     =                       bsize=4096   blocks=524288, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

Step 5: Mount the **/app** directory with **/dev/vg-1/lv-1** again and check the size of the directory using following command.

```
mount /dev/vg-1/lv-1 /app/

df -h /app
```

```
[root@redhat ~]# mount /dev/vg-1/lv-1 /app/
[root@redhat ~]# df -h /app
Filesystem               Size  Used Avail Use% Mounted on
/dev/mapper/vg--1-lv--1  2.0G   47M  2.0G   3% /app
```

As you can see, **/app** directory size has been reduced to 2GB.

Step 6: Restore the file system dump backup (**/tmp/app.dump**) to **/app** directory using following command.

```
xfsrestore -f /tmp/app.dump /app
```

```
[root@redhat ~]# xfsrestore -f /tmp/app.dump /app
xfsrestore: using file dump (drive_simple) strategy
xfsrestore: version 3.1.10 (dump format 3.0) - type ^C for status and control
xfsrestore: searching media for dump
xfsrestore: examining media file 0
xfsrestore: dump description:
xfsrestore: hostname: redhat
xfsrestore: mount point: /app
xfsrestore: volume: /dev/mapper/vg--1-lv--1
xfsrestore: session time: Thu Jan 19 12:22:01 2023
xfsrestore: level: 0
xfsrestore: session label: ""app_dir_backup""
xfsrestore: media label: "app-dir-data"
xfsrestore: file system id: 6a129e6c-e194-4dbb-86e5-1f1c104c43f3
xfsrestore: session id: 0009118b-438a-4f68-9912-d0a127e2b095
xfsrestore: media id: 737ef897-0205-43ee-b3a0-6f1cbcdf40a1
xfsrestore: using online session inventory
xfsrestore: searching media for directory dump
xfsrestore: reading directories
xfsrestore: 1 directories and 11 entries processed
xfsrestore: directory post-processing
xfsrestore: restoring non-directory files
xfsrestore: restore complete: 0 seconds elapsed
xfsrestore: Restore Summary:
xfsrestore:   stream 0 /tmp/app.dump OK (success)
xfsrestore: Restore Status: SUCCESS
```

Now you can check the files are available in **/app** directory with **ls** command.

```
ls -l /app
```

--End of Chapter 10.

Chapter 11: Linux Disk Partition Management – Debian Systems

11.1 How to Create a new partition in Debian Distros:

Step 1: Add a disk into your ubuntu machine and verify disk is added. I have added /dev/sdb disk for this example.

The following command list **/dev/sdb** disk details.

```
fdisk -l /dev/sdb
```

```
root@tn:~# fdisk -l /dev/sdb
Disk /dev/sdb: 2.1 GiB, 2158220800 bytes, 4215275 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xca9bd40e
```

Step 2: Create new Partition.

The following command will open **fdisk** console for **/dev/sdb** disk.

```
fdisk /dev/sdb
```

Type **n** to create new partition,

```
n
```

```
root@tn:~# fdisk /dev/sdb

Welcome to fdisk (util-linux 2.34).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): n
```

Specify the type of partition using the **p** for primary and **e** for extended. This will create a primary partition.

```
Partition type: p
```

```
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
```

The console will prompt for the number to be given to the partition. We can give any number from 1 to 4. I am giving 1 as a partition number because this is the first partition for **/dev/sdb** disk.

```
1
```

Specify the size of the partition (First sector & Last sector),

```
Example:
First sector: 2048
Last sector:
[Note: If you didn't mention Last sector size, it will allocate all free space to
this partition]
```

Pressing **enter** will create our 1st partition successfully with 2GB size.

write the changes to the disk using the **w** command, else use the **q** command to quit without writing.

```
w
```

```
root@tn:~# fdisk /dev/sdb

Welcome to fdisk (util-linux 2.34).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-4215274, default 2048): 2048
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-4215274, default 4215274):

Created a new partition 1 of type 'Linux' and of size 2 GiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

Step 3: Create file system for new partition. In order to specify the file system to be used in each partition,

we can use the **mkfs (make file system)** command.

The following example we are creating **ext4** file system.

```
mkfs.ext4 /dev/sdb1
```

```
root@tn:~# mkfs.ext4 /dev/sdb1
mke2fs 1.45.5 (07-Jan-2020)
Creating filesystem with 526653 4k blocks and 131920 inodes
Filesystem UUID: 1143bfc9-0ae8-45be-afb1-46a2558469a6
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

Step 4: Mount the file system.

Create new directory.

```
mkdir /database

# Mount /dev/sdb1 to /database directory.
mount /dev/sdb1 /database
```

```
root@tn:~# mkdir /database
root@tn:~# mount /dev/sdb1 /database
```

Check the mounting.

```
df -h /database
```

```
root@tn:~# df -h /database
Filesystem      Size  Used Avail Use% Mounted on
/dev/sdb1       2.0G   24K  1.8G   1% /database
```

Step 5: Mount the file system permanently.

All these mounting are temporary by default. Once we reboot the system, mounting will be reverted. To make it permanent, we must edit the File System Table of the Operating System.

```
Vim /etc/fstab
```

```
root@tn:~# vim /etc/fstab
```

[**Note:** A small error in this file can cause the system to be unbootable and can make the entire system to be useless. So edit carefully]

Add our mounted file systems details in the **/etc/fstab** file & save the file,

```
/dev/sdb1   /database   ext4   defaults 0 0
```

```
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
/dev/mapper/vgubuntu-root /               ext4    errors=remount-ro 0       1
# /boot/efi was on /dev/sda1 during installation
UUID=B775-79EA  /boot/efi       vfat    umask=0077      0       1
/dev/mapper/vgubuntu-swap_1 none            swap    sw              0       0
/dev/sdb1       /database       ext4    defaults                0       0
```

11.2 How to delete a partition in Debian Distros:

In this example, we are going to delete **/database** mount point.

/database is mounted with **/dev/sdb** physical disk.

Step 1: First unmount the file system.

The following command will unmount the **/app** partition in your linux system.

```
umount /database
```

```
[root@redhat ~]# umount /app
[root@redhat ~]# df -h
Filesystem              Size  Used Avail Use% Mounted on
devtmpfs                4.0M     0  4.0M   0% /dev
tmpfs                   886M     0  886M   0% /dev/shm
tmpfs                   355M  5.0M  350M   2% /run
/dev/mapper/rhel-root    16G  1.3G   15G   8% /
/dev/sda1               471M  270M  202M  58% /boot
tmpfs                   178M     0  178M   0% /run/user/0
```

Step 2: Remove the file system entry from **/etc/fstab** file.

Delete the following line from **/etc/fstab** file.

```
/dev/sdb1  /database  ext4  defaults 0 0
```

```
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
/dev/mapper/vgubuntu-root /               ext4    errors=remount-ro 0       1
# /boot/efi was on /dev/sda1 during installation
UUID=B775-79EA  /boot/efi       vfat    umask=0077      0       1
/dev/mapper/vgubuntu-swap_1 none            swap    sw              0       0
/dev/sdb1       /database       ext4    defaults                0       0
~
~
```

Step 3: Delete the partition.

The following command will open the **fdisk** console for **/dev/sdb** disk.

```
fdisk /dev/sdb

# To delete a partition type,
```

```
d

# It will prompt the partition number you want to delete(type the number). In this
example partition no 1 only available and selected by default.
1

# Write the changes to the disk using the w command, else use the q command to quit
without writing.
w
```

```
root@tn:~# fdisk /dev/sdb

Welcome to fdisk (util-linux 2.34).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): d
Selected partition 1
Partition 1 has been deleted.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

11.3 How to Create partition using LVM in Debian Distros:

Step 1: Add a disk into your Ubuntu machine and verify disk is added. I have added **/dev/sdb** disk for this example.

The following command list **/dev/sdb** disk details.

```
fdisk -l /dev/sdb
```

```
root@tn:~# fdisk -l /dev/sdb
Disk /dev/sdb: 2.1 GiB, 2158220800 bytes, 4215275 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xca9bd40e
```

Step 2: Create Physical volume(PV) with **pvcreate** command.

```
pvcreate /dev/sdb
```

[**Note:** Replace **/dev/sdb** with appropriate partition name.]

```
root@tn:~# pvcreate /dev/sdb
  Physical volume "/dev/sdb"  successfully created.
```

Verify the physical volume **/dev/sdb** using following command.

```
Pvdisplay /dev/sdb
```

```
root@tn:~# pvdisplay /dev/sdb
  "/dev/sdb " is a new physical volume of "<2.01 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/sdb
  VG Name
  PV Size               <2.01 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               9G3o4b-nxT0-maEq-gDuZ-msTN-I0nI-7wK8oB
```

Step 3: Create Volume Group (VG) with **vgcreate** command. In this example I am creating volume group named **vg-1.**

```
vgcreate vg-1 /dev/sdb
```

[**Note:** Replace **/dev/sdb** with appropriate pv name.]

```
root@tn:~# vgcreate vg-1 /dev/sdb
  Volume group "vg-1" successfully created
```

Verify the Volume group **vg-1**.

```
vgdisplay vg-1
```

```
root@tn:~# vgdisplay vg-1
  --- Volume group ---
  VG Name               vg-1
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               <2.01 GiB
  PE Size               4.00 MiB
  Total PE              514
  Alloc PE / Size       0 / 0
  Free  PE / Size       514 / <2.01 GiB
  VG UUID               fjdiC7-DmAF-TKiG-1FMB-eCUl-TqES-uZZele
```

Step 4: Create Logical Volume (LV) with **lvcreate** command. It will create logical volume named **lv-1** with 1.99G**.**

```
lvcreate -L 2G -n lv-1 vg-1
```

```
root@tn:~# lvcreate -L 2G -n lv-1 vg-1
  Logical volume "lv-1" created.
```

Verify the Logical volume **lv-1**.

```
lvdisplay /dev/vg-1/lv-1
```

```
root@tn:~# lvdisplay /dev/vg-1/lv-1
  --- Logical volume ---
  LV Path                /dev/vg-1/lv-1
  LV Name                lv-1
  VG Name                vg-1
  LV UUID                IrcOyz-mMN7-4MlE-Dduv-pvbF-nOGr-dlYojA
  LV Write Access        read/write
  LV Creation host, time tn, 2023-01-19 10:41:13 +0530
  LV Status              available
  # open                 0
  LV Size                2.00 GiB
  Current LE             512
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:2
```

Step 5: Create file system for new lvm partition

The following example we are creating **ext4** file system.

```
mkfs.ext4 /dev/vg-1/lv-1
```

```
root@tn:~# mkfs.ext4 /dev/vg-1/lv-1
mke2fs 1.45.5 (07-Jan-2020)
Creating filesystem with 524288 4k blocks and 131072 inodes
Filesystem UUID: 1694d8de-1db6-4353-af5e-63c3bf79eb3f
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

Step 6: Mount the file system.

Create new directory.

```
mkdir /database
```

Mount **/dev/vg-1/lv-1** to **/app** directory.

```
mount /dev/vg-1/lv-1 /database
```

Check the mounting.

```
df -h
```

```
root@tn:~# mount /dev/vg-1/lv-1 /database
root@tn:~# df -h /database
Filesystem              Size  Used Avail Use% Mounted on
/dev/mapper/vg--1-lv--1  2.0G   24K  1.8G   1% /database
```

Step 7: Mount the file system permanently.

All these mounting are temporary by default. Once we reboot the system, mounting will be reverted. To make it permanent, we must edit the File System Table of the Operating System.

```
vim /etc/fstab
```

```
root@tn:~# vim /etc/fstab
```

[**Note:** A small error in this file can cause the system to be unbootable and can make the entire system to be useless. So edit carefully]

Add our mounted file systems details in the **/etc/fstab** file & save the file,

```
/dev/vg-1/lv-1   /database   ext4   defaults 0 0
```

```
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
/dev/mapper/vgubuntu-root /               ext4    errors=remount-ro 0       1
# /boot/efi was on /dev/sda1 during installation
UUID=B775-79EA  /boot/efi       vfat    umask=0077      0       1
/dev/mapper/vgubuntu-swap_1 none            swap    sw              0       0
/dev/vg-1/lv-1  /database       ext4    defaults        0       0
~
```

11.4 How to Extend partition size using LVM in Debian Distros:

In this example, we are going to increase the size of **/database** mount point size from 2GB to 5GB.

/database mounted with **/dev/vg-1/lv-1** (VG Name – vg-1, LV Name – lv-1).

Step 1: Add a disk into your ubuntu machine and verify disk is added. I have added additional **/dev/sdc** disk for this example.

The following command list **/dev/sdc** disk details.

```
fdisk -l /dev/sdc
```

Step 2: Create Physical volume (PV) with **pvcreate** command.

```
pvcreate /dev/sdc
```

[**Note:** Replace **/dev/sdc** with appropriate partition name.]

```
root@tn:~# pvcreate /dev/sdc
  Physical volume "/dev/sdc"  successfully created.
```

Verify the physical volume **/dev/sdc** using following command.

```
pvdisplay /dev/sdc
```

```
root@tn:~# pvdisplay /dev/sdc
  "/dev/sdc " is a new physical volume of "<3.01 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/sdc
  VG Name
  PV Size               <3.01 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               xrfnp4-5yrb-onUN-z2nT-basy-e4cH-8XQVKg
```

Step 3: Extend the volume group **vg-1**.

```
vgextend vg-1 /dev/sdc
```

[**Note:** Replace VG Name **vg-1** & PV Name **/dev/sdc** with appropriate VG & PV name.]

As you can see, Volume Group **vg-1** size has been extended to 5GB.

```
root@tn:~# vgextend vg-1 /dev/sdc
  Volume group "vg-1" successfully extended
root@tn:~#
root@tn:~# vgs
  VG       #PV #LV #SN Attr   VSize   VFree
  vg-1       2   1   0 wz--n-  <5.02g <3.02g
  vgubuntu   1   2   0 wz--n- <19.50g 36.00m
```

Step 4: Now Extend the logical volume.

```
lvextend -L +3G /dev/vg-1/lv-1
```

[**Note:** Replace LV Name **/dev/vg-1/lv-1** with appropriate lv name.]

```
root@tn:~# lvextend -L +3G /dev/vg-1/lv-1
  Size of logical volume vg-1/lv-1 changed from 2.00 GiB (512 extents) to 5.00 GiB (1280 extents).
  Logical volume vg-1/lv-1 successfully resized.
root@tn:~#
root@tn:~# lvs
  LV     VG       Attr       LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  lv-1   vg-1     -wi-ao----   5.00g
```

Step 5: Resize the file system.

```
resize2fs /dev/vg-1/lv-1
```

[**Note:** Replace LV Name **/dev/vg-1/lv-1** with appropriate lv name.]

```
root@tn:~# resize2fs /dev/vg-1/lv-1
resize2fs 1.45.5 (07-Jan-2020)
Filesystem at /dev/vg-1/lv-1 is mounted on /database; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 1
The filesystem on /dev/vg-1/lv-1 is now 1310720 (4k) blocks long.
```

Now check the **/database** mount directory size with **df** command.

```
df -h /database
```

```
root@tn:~# df -h /database
Filesystem              Size  Used Avail Use% Mounted on
/dev/mapper/vg--1-lv--1  4.9G   24K  4.7G   1% /database
```

11.5 How to reduce LVM partition size in Debian Distros:

Here we are going to reduce 3GB size of the mount point **/database** directory,

/database is mounted with **/dev/vg-1/lv-1** & having **ext4** filesystem.

Here **lv-1** is the logical volume & **vg-1** is the Volume Group.

Before reducing the lvm size, it is always good to backup the data, so that it will not be a data loss if something goes wrong.

To Reduce a logical volume there are 5 steps needed to be done very carefully.

Step 1: Unmount the mount point directory. The following example we are reducing **/app** directory.

```
umount /database
```

```
root@tn:~# umount /database
root@tn:~#
```

Step 2: Verify the filesystem errors using following command. It must pass all 5 checks.

```
e2fsck -f /dev/vg-1/lv-1
```

```
root@tn:~# e2fsck -f /dev/vg-1/lv-1
e2fsck 1.45.5 (07-Jan-2020)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/vg-1/lv-1: 11/327680 files (0.0% non-contiguous), 39006/1310720 blocks
```

Step 3: Reduce the file system size to 2GB.

[**Note: /database** has total **5GB** size, in this example we are reducing 3 GB. So after 3GB reduced, **/databse** directory size will be **2GB**.]

```
resize2fs /dev/vg-1/lv-1 2GB
```

```
root@tn:~# resize2fs /dev/vg-1/lv-1 2G
resize2fs 1.45.5 (07-Jan-2020)
Resizing the filesystem on /dev/vg-1/lv-1 to 524288 (4k) blocks.
The filesystem on /dev/vg-1/lv-1 is now 524288 (4k) blocks long.
```

Step 4: Now reduce the logical volume size using following command.

```
lvreduce -L -3G /dev/vg-1/lv-1
```

```
root@tn:~# lvreduce -L -3G /dev/vg-1/lv-1
  WARNING: Reducing active logical volume to 2.00 GiB.
  THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce vg-1/lv-1? [y/n]: y
  Size of logical volume vg-1/lv-1 changed from 5.00 GiB (1280 extents) to 2.00 GiB (512 extents).
  Logical volume vg-1/lv-1 successfully resized.
```

Step 5: Mount the **/app** directory with **/dev/vg-1/lv-1** again using following command and check the directory size using **df** command.

```
mount /dev/vg-1/lv-1 /database

df -h /database
```

```
root@tn:~# mount /dev/vg-1/lv-1 /database
root@tn:~# df -h /database/
Filesystem               Size  Used Avail Use% Mounted on
/dev/mapper/vg--1-lv--1  2.0G   24K  1.8G   1% /database
```

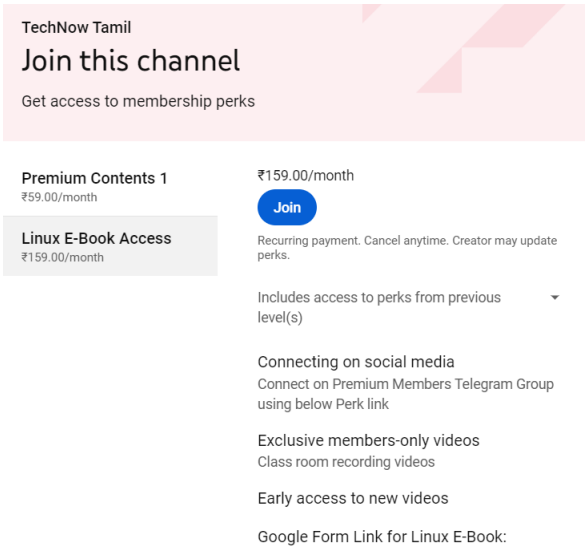
--End of Chapter 11.

Get Premium Version of this Book:

Hi All,

I hope you're enjoying this E-Book. There is also a Premium version available, featuring additional chapters and topics. You can access these extra benefits by joining my YouTube membership plan along with this Premium Guide.

Click Here to Join Technow Tamil Membership and Get Linux Admin Guide Premium E-Book with additional perks.



Key Features:

- Pay once to access all future versions of this e-book—no monthly fees.
- Additionally, you'll gain access to my upcoming books, including PostgreSQL, MySQL, Apache Webserver, and Nginx Webserver Admin Guide Premium E-Books. You will automatically receive access to these e-books upon their launch.

Channel Membership All Perks:

1) Linux Admin Guide Premium E-Book Access
2) Telegram Premium Group Access
3) Exclusive members-only Premium videos
4) Upcoming E-Books Access (PostgreSQL, MySQL, Apache Webserver, Nginx Webserver Admin Guides)

--- End Of Page---