to create the folder

#mkdir foldername

To go inside the folder

#cd foldername

to create file

#vi filename

# nano filename

# vim finename

enter the detials

esc :wqx`

remove commands:

to remove the file

# rm filename

To remove the folder/directory

#rm -rf

rm: This is the command to remove files or directories.

-r: This flag stands for "recursive," meaning it will delete directories and their contents.

-f: This stands for "force," meaning it will delete files without prompting for confirmation.

#rm -rf \* --> removes all the files and directories(does not used in company)

# rm fileName --> removes the file frm the directory

#pwd --> present working directory

list commands:

#ls --> list the files in the directory

root@ip-172-31-42-146:~# ls -l

total 24

-rw-r--r-- 1 (if starts from -rw than its a file)

drwxr-xr-x 3 (if starts from d than its a folder)

#ls -a : The ls -a command in Unix-based systems lists all files and directories in the current directory, including hidden ones.

#ls -l :

The ls -l command in Unix-based systems provides a detailed (long) listing of files and directories in the current directory. The output includes more information than the basic ls command. Here's a breakdown of the information typically displayed:

File Type and Permissions: The first column shows the file type and permissions. For example, -rwxr-xr-- indicates a regular file with specific read, write, and execute permissions.

Number of Links: The second column shows the number of hard links to the file or directory.

Owner: The third column displays the username of the file's owner.

Group: The fourth column shows the group name associated with the file.

File Size: The fifth column shows the size of the file in bytes.

Modification Date and Time: The sixth column shows the date and time when the file was last modified.

File/Directory Name: The final column displays the name of the file or directory.

#ls -la

The command ls -la is a combination of ls -l and ls -a, which gives you a long-format listing of all files and directories, including hidden ones. Here’s what each part does:

ls: Lists files and directories.

-l: Provides a detailed (long) format listing.

-a: Includes hidden files (those starting with a dot).

#ls -lt -->sorts the output by modification time, with the most recently modified files appearing first.

#ls -ltr ---> reverses the order of the sort, so the oldest files are listed first.

ls: Lists files and directories.

-l: Uses the long format to show detailed information (permissions, number of links, owner, group, size, and modification date).

-t: Sorts by modification time, with the most recently modified files appearing first.

-r: Reverses the order of the sort, so the oldest files are listed first.

CAT COMMANDS:

#cat file\_name -- displays all the content in the files

cat -n file\_name: Displays the contents of the file with line numbers.

cat -b file\_name: Displays the contents with line numbers only for non-blank lines.

cat file1 file2: Concatenates and displays the contents of multiple files in sequence.

COPY COMMANDS

COPY command cp are used to copy file from source path to distination path

cp file1 folder1

here file1 is source path and folder1 id distination path

file1 will be copied to folder1

The command cp -r folder1 folder2 is used to copy the contents of folder1 into folder2 recursively. Here’s a breakdown of the command:

cp: The command for copying files and directories.

-r: This flag stands for "recursive," meaning it will copy directories and their contents, including all subdirectories and files.

MOVE COMMAND :

MOVE COMMAND acts like cut and paste

# mv folder1 folder2

here folder1 is the source\_path and folder2 is the distination path

Additional Options

-i: Prompts for confirmation before overwriting existing files or directories.

-u: Moves only when the source is newer than the destination or when the destination is missing.

-v: Verbose mode, which shows the files and directories being moved.

using move commands we can also rename the file

mv filename naveen

TAIL AND HEAD COMMANDS:

HOW TO CHECK FIRSTTWO LINES OF FILE

using head commands: # head -2 filename

HOW TO CHECK LASTTWO LINES OF THE FILE

using tail commands: # tail -2 filename

TO CHECK THE COUNT OF THE FILE:

#wc file\_name

op: 10 100 500 file\_name

10: Number of lines

100: Number of words

500: Number of bytes (or characters)

file\_name: The name of the file

#wc -c file\_name --> displays only character

#wc -w file\_name displays only words

#wc -l file\_name displays only line count

HOW TO DISPLAY THE SIZE OF THE FOLDER:

#du -sh

The du command in Unix-based systems is used to estimate file and directory space usage. The -s and -h options

are commonly used together to provide a summary of the total disk usage of a directory in a human-readable format.

Here’s a breakdown of the options:

-s: Stands for "summarize." It provides the total size of the specified directory or file, without listing the sizes of individual subdirectories or files.

-h: Stands for "human-readable." It formats the output in a way that's easier to read, using units like KB, MB, GB, etc.

#free

The free command in Unix-based systems is used to display information about the system's memory usage.

It provides a summary of the total, used, free, and available memory, including both physical RAM and swap space.

op:

total used free shared buff/cache available

Mem: 16384228 6548972 2236576 239404 7599680 9304212

Swap: 2097148 0 2097148

# free -h ---> provides information in human readable format

op:

total used free shared buff/cache available

Mem: 16G 6.3G 2.1G 234M 7.3G 8.8G

Swap: 2.0G 0B 2.0G

file permission

r--> read r=4

w--> write w=2

x--> execute x=1

user group others

rwx rwx rwx

421 421 421

7 7 7

#chmod 777 file\_name

all permissions i.e, read, write and execute permissions given all user groups and others.

The chmod command in Unix-based systems is used to change the file mode

(permissions) of a file or directory. The 777 in chmod 777 file\_name sets

the permissions for the file or directory to be fully accessible by everyone. Here’s a detailed explanation:

Breakdown of chmod 777

chmod: The command to change file permissions.

777: The permission settings in numeric (octal) format.

7: Represents read (4), write (2), and execute (1) permissions.

Each digit corresponds to a set of permissions:

First digit (7): Permissions for the file owner.

Second digit (7): Permissions for the group.

Third digit (7): Permissions for others (everyone else).

Numeric Permission Values

4: Read permission

2: Write permission

1: Execute permission

The sum of these values gives the permission level:

7: Read (4) + Write (2) + Execute (1)

Example Usage

If you run:

bash

Copy code

chmod 777 file\_name

It sets the following permissions:

Owner: Read, write, and execute.

Group: Read, write, and execute.

Others: Read, write, and execute.

Example Permissions:

sql

Copy code

-rwxrwxrwx 1 user group 1234 Sep 9 14:22 file\_name

In this example:

rwx (for owner): Read, write, and execute permissions.

rwx (for group): Read, write, and execute permissions.

rwx (for others): Read, write, and execute permissions.

Security Implications

Setting permissions to 777 makes the file or directory fully accessible to anyone, which can pose a significant security risk:

Read: Any user can view the file’s content.

Write: Any user can modify or delete the file.

Execute: Any user can execute the file if it is a script or binary.

It is generally advised to avoid 777 permissions unless absolutely necessary. Instead, use more restrictive permissions that follow the principle of least privilege, granting only the access required.

Other Common chmod Permissions

755: Read, write, and execute for the owner; read and execute for group and others.

644: Read and write for the owner; read-only for group and others.

700: Read, write, and execute for the owner; no permissions for group and others.

If you have more questions or need further assistance, feel free to ask!

chmod u+x file\_name

Add execute permission for the owner of the file:

chmod u+x filename

Explanation:

u: Refers to the user (owner) of the file.

+x: Adds execute permission.

Add execute permission for the group:

chmod g+x filename

Explanation:

g: Refers to the group associated with the file.

+x: Adds execute permission.

Add execute permission for others:

chmod o+x filename

Explanation:

o: Refers to others (everyone else).

+x: Adds execute permission.

FIND COMMAND:

find is the command to find particular files

find . -type f -iname file\_name

find is the command to find particular folder

find . -type m -iname file\_name

command to find file which have created 3months ago?

find . -type f -mtime +90

ago means +

command to find file which have created within 3months?

find . -type f -mtime -90

command to find file which was created within 60 days and 30 ago?

find . -type f -mtime -60 ! -mtime +30

command to find the file which was created within 10 minutes?

find . -type f -mmin -10

command to find the file which was created 10 minutes ago?

find . -type f -mmin +10

command to find the file with full permission

find . -type f -perm 777

command to find the empty file

find . -type f -empty

command to find the file size lessthan 4kb

find . -type f -size -4k

command to find the file size greaterthan 4kb

find . -type f -size +4k

links:

link is a shortcut to a file meanwhile if we make many changes in the file changes

will happen in link also viseversa

softlink: softlink is a shortcut to a file if we make many changes to the file

it will reflect in link, if the file is deleted and link does not work.

# ln -s filename link\_name

(link softlink filename link\_name)

note: if file is delete then link will be displayed in RED.

hardlink: hardlink is a shortcut to a file if we make many changes to the file

it will reflect in the link, if the file is deleted than also link will work.

USER ADD:

how to add user:

#useradd user\_name

To set password:

#passwd user\_name

enter password--> password authenticated successfully.

TO SWICHT TO USER

#su user\_name

#bash

How to check whether user is added or not?

cat /etc/group

added user will be displayed at last.

USER GROUP:

command to add group

#groupadd devops

command to add user ansible in group docker

usermod -aG docker ansible

Explanation:

The command you're referencing is used to add a user to the Docker group on a Unix-like operating system, which allows that user to run Docker commands without needing superuser privileges.

Here’s a breakdown of the command:

usermod: This is a command used to modify a user account.

-aG: These options are used together to append the user to a group.

-a stands for append, and -G specifies

the group to which the user is being added.

docker: This is the name of the group to which the user is being added.

ansible: This is the username being added to the Docker group.

So, usermod -aG docker ansible will add the user ansible to

the docker group. After running this command, you might need to

log out and log back in for the group membership changes to take effect.

to check command:

#getent group docker

or # cat /etc/group

output: docker:x:1001:ansible

command to delete user from group:

# gpasswd -d ansible docker

command to delete group:

#groupdel group\_name

command to delete user:

#userdel user\_name

FILE COMPRESSING COMMANDS:

WE HAVE THREE TYPES OF FILES:

1. TAR

2. ZIP

3.GZIP

command to commpress tarfile

tar -cvf folder\_name.tar filename1 filename2

command to extract the tarfile

tar -xvf folder\_name.tar

command to commpress zip\_file

#zip folder.zip file1 file2 file1

file1 file2 will be added in folder.zip

command to unzip the file:

#unzip folder.zip

#ls

file1 file2 file3

GZIP COMPRESS ONLY SINGLE FILE WITH EXTENSION.gz

#gzip file1

ls

file1.gs

to extact tar-gzip file:

tar -zxvf jsp.tar.gz

command to display ip address of server:

#hostname -i

command to display server name:

#hostname

command to display server full name:

#hostname -f

command to display system detais:

#uname -a

command to display os versiion names:

#uname -v