rm file-nme <-- to delete a file wwith prompt with prompt

syntax:

rm file-nme

example:

[root@jenkins opt]# ll

total 0

-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen10

-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen2

-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen3

-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen4

-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen5

-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen6

-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen7

-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen8

-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen9

[root@jenkins opt]# rm cloudgen2

rm: remove regular empty file 'cloudgen2'? y

[root@jenkins opt]# ll

total 0

-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen10

-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen3

-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen4

-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen5

-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen6

-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen7

-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen8

-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen9

[root@jenkins opt]#

rm file-nme file-nme file-nme <-- to delete multiple files with prompt

[root@jenkins opt]# ll

total 0

-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen10

-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen3

-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen4

-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen5

-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen6

-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen7

-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen8

-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen9

[root@jenkins opt]# rm cloudgen3 cloudgen4 cloudgen7

rm: remove regular empty file 'cloudgen3'? y

rm: remove regular empty file 'cloudgen4'? y

rm: remove regular empty file 'cloudgen7'? y

```
[root@jenkins opt]# ll
total 0
-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen10
-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen5
-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen6
-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen8
-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen9
[root@jenkins opt]#
```

To delete a file or files without prompt.

to delete single file

syntax rm -f file-nme

example:

```
[root@jenkins opt]# ll
total 0
-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen10
-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen5
-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen6
-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen8
-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen9
```

```
[root@jenkins opt]# rm -f cloudgen5
[root@jenkins opt]# ll
total 0
-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen10
-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen6
-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen8
-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen9
[root@jenkins opt]#
```

To delete multiple files without prompt.>!

syntax rm -f file-nme file-nme file-nme.. nth file-nme

example:

```
[root@jenkins opt]# ll
total 0
-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen10
-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen6
-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen8
-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen9

[root@jenkins opt]#
[root@jenkins opt]# rm -f cloudgen9 cloudgen8 cloudgen6
[root@jenkins opt]# ll
```

total 0

-rw-r--r--. 1 root root 0 Apr 29 10:17 cloudgen10

[root@jenkins opt]#

to Delete All Files at a time:

example:

rm -f * <-- by this command we can delete all files at a time from present working directory.

Working With directory

To Create a directory

syntax:

mkdir directory-name or Directory-Name (lower caser or Upper Case)

example:

mkdir CloudGen

mkdir ravi

TO Create multiple Directories at a time.

mkdir python oralce mysql MySQL DevOps

To Create multiple Directories with Series.

mkdir ravi{1..5} raja{1..5}

****

Directories creation in remote-path or in target-path path

mkdir directory-name <-- we are creating a directory in present working directory

[root@jenkins opt]# mkdir cloud

[root@jenkins opt]# pwd

/opt

[root@jenkins opt]# ll

total 0

-rw-r--r--. 1 root root 0 Apr 29 11:13 a

-rw-r--r--. 1 root root 0 Apr 29 11:13 b

-rw-r--r--. 1 root root 0 Apr 29 11:13 c

drwxr-xr-x. 2 root root 6 Apr 29 11:18 cloud

-rw-r--r--. 1 root root 0 Apr 29 11:13 d

mkdir directory-name <target-path/target-directory/destination-path>

for example:

[root@jenkins opt]# ll /root/

total 4

-rw-------. 1 root root 1034 Mar 16 10:23 anaconda-ks.cfg

[root@jenkins opt]#

[root@jenkins opt]# mkdir /root/cloud

[root@jenkins opt]# ll /root/

total 4

-rw-------. 1 root root 1034 Mar 16 10:23 anaconda-ks.cfg

drwxr-xr-x. 2 root root 6 Apr 29 11:18 cloud

[root@jenkins opt]#

TO Delete directory.

syntax: with prompt:

rm -r directory-name

for example:

[root@jenkins opt]# ll

total 0

drwxr-xr-x. 2 root root 6 Apr 29 10:41 CloudGen

drwxr-xr-x. 2 root root 6 Apr 29 10:44 ravi4

[root@jenkins opt]# rm -r ravi4

rm: remove directory 'ravi4'? y

[root@jenkins opt]#

TO Delete directory.

syntax: with out prompt:

rm -r directory-name

for example:

```
[root@jenkins opt]# ll
total 0
drwxr-xr-x. 2 root root 6 Apr 29 10:41 CloudGen
drwxr-xr-x. 2 root root 6 Apr 29 10:42 DevOps
drwxr-xr-x. 2 root root 6 Apr 29 10:44 ravi3

[root@jenkins opt]# rm -rf ravi3

[root@jenkins opt]# ll
total 0
drwxr-xr-x. 2 root root 6 Apr 29 10:41 CloudGen
drwxr-xr-x. 2 root root 6 Apr 29 10:42 DevOps

[root@jenkins opt]#
```

to delete multiple Directories at a time:

example:

```
[root@jenkins opt]# ll
total 0
drwxr-xr-x. 2 root root 6 Apr 29 10:41 CloudGen
drwxr-xr-x. 2 root root 6 Apr 29 10:42 DevOps
drwxr-xr-x. 2 root root 6 Apr 29 10:42 mysql
drwxr-xr-x. 2 root root 6 Apr 29 10:42 MySQL
drwxr-xr-x. 2 root root 6 Apr 29 10:42 oralce
drwxr-xr-x. 2 root root 6 Apr 29 10:42 python
drwxr-xr-x. 2 root root 6 Apr 29 10:44 raja1
drwxr-xr-x. 2 root root 6 Apr 29 10:44 raja2
drwxr-xr-x. 2 root root 6 Apr 29 10:44 raja3
drwxr-xr-x. 2 root root 6 Apr 29 10:44 raja4
drwxr-xr-x. 2 root root 6 Apr 29 10:44 raja5
drwxr-xr-x. 2 root root 6 Apr 29 10:41 ravi
drwxr-xr-x. 2 root root 6 Apr 29 10:44 ravi1
drwxr-xr-x. 2 root root 6 Apr 29 10:44 ravi2

[root@jenkins opt]# rm -rf  MySQL ravi
[root@jenkins opt]# ll
total 0
drwxr-xr-x. 2 root root 6 Apr 29 10:41 CloudGen
drwxr-xr-x. 2 root root 6 Apr 29 10:42 DevOps
drwxr-xr-x. 2 root root 6 Apr 29 10:42 mysql
drwxr-xr-x. 2 root root 6 Apr 29 10:42 oralce
```

drwxr-xr-x. 2 root root 6 Apr 29 10:42 python

drwxr-xr-x. 2 root root 6 Apr 29 10:44 raja1

drwxr-xr-x. 2 root root 6 Apr 29 10:44 raja2

drwxr-xr-x. 2 root root 6 Apr 29 10:44 raja3

drwxr-xr-x. 2 root root 6 Apr 29 10:44 raja4

drwxr-xr-x. 2 root root 6 Apr 29 10:44 raja5

drwxr-xr-x. 2 root root 6 Apr 29 10:44 ravi1

drwxr-xr-x. 2 root root 6 Apr 29 10:44 ravi2

[root@jenkins opt]# rm -rf raja5 raja3

[root@jenkins opt]# ll

total 0

drwxr-xr-x. 2 root root 6 Apr 29 10:41 CloudGen

drwxr-xr-x. 2 root root 6 Apr 29 10:42 DevOps

drwxr-xr-x. 2 root root 6 Apr 29 10:42 mysql

drwxr-xr-x. 2 root root 6 Apr 29 10:42 oralce

drwxr-xr-x. 2 root root 6 Apr 29 10:42 python

drwxr-xr-x. 2 root root 6 Apr 29 10:44 raja1

drwxr-xr-x. 2 root root 6 Apr 29 10:44 raja2

drwxr-xr-x. 2 root root 6 Apr 29 10:44 raja4

drwxr-xr-x. 2 root root 6 Apr 29 10:44 ravi1

drwxr-xr-x. 2 root root 6 Apr 29 10:44 ravi2

[root@jenkins opt]#

example:

rm -rf file-nme{starting-number .. ending_number }

for example:

[root@jenkins opt]# ll

total 0

drwxr-xr-x. 2 root root 6 Apr 29 10:41 CloudGen

drwxr-xr-x. 2 root root 6 Apr 29 10:42 DevOps

drwxr-xr-x. 2 root root 6 Apr 29 10:42 mysql

drwxr-xr-x. 2 root root 6 Apr 29 10:42 oralce

drwxr-xr-x. 2 root root 6 Apr 29 10:42 python

drwxr-xr-x. 2 root root 6 Apr 29 10:44 raja1

drwxr-xr-x. 2 root root 6 Apr 29 10:44 raja2

drwxr-xr-x. 2 root root 6 Apr 29 10:44 raja4

drwxr-xr-x. 2 root root 6 Apr 29 10:44 ravi1

drwxr-xr-x. 2 root root 6 Apr 29 10:44 ravi2

[root@jenkins opt]# rm -rf raja{1..4}

[root@jenkins opt]# ll

total 0

drwxr-xr-x. 2 root root 6 Apr 29 10:41 CloudGen

drwxr-xr-x. 2 root root 6 Apr 29 10:42 DevOps

drwxr-xr-x. 2 root root 6 Apr 29 10:42 mysql

drwxr-xr-x. 2 root root 6 Apr 29 10:42 oralce

drwxr-xr-x. 2 root root 6 Apr 29 10:42 python

drwxr-xr-x. 2 root root 6 Apr 29 10:44 ravi1

drwxr-xr-x. 2 root root 6 Apr 29 10:44 ravi2

[root@jenkins opt]#

---

mv <-- this is a command and it used for to rename file/directory

syntax: to file rename.

mv old/existed-file new-file-name

syntax: to directory rename.

mv old/existed-directory-name new-directory-name

for example:

[root@jenkins opt]# ll

total 0

drwxr-xr-x. 2 root root 6 Apr 29 10:41 CloudGen

drwxr-xr-x. 2 root root 6 Apr 29 10:42 DevOps

-rw-r--r--. 1 root root 0 Apr 29 11:00 java

[root@jenkins opt]# mv java java17

```
[root@jenkins opt]# ll
total 0
drwxr-xr-x. 2 root root 6 Apr 29 10:41 CloudGen
drwxr-xr-x. 2 root root 6 Apr 29 10:42 DevOps
-rw-r--r--. 1 root root 0 Apr 29 11:00 java17

[root@jenkins opt]#
```

TO Rename directory:

for example:

```
[root@jenkins opt]# ll
total 0
drwxr-xr-x. 2 root root 6 Apr 29 10:41 CloudGen
drwxr-xr-x. 2 root root 6 Apr 29 10:42 DevOps
-rw-r--r--. 1 root root 0 Apr 29 11:00 java17

[root@jenkins opt]# mv CloudGen/ Cloud

[root@jenkins opt]# ll
total 0
```

drwxr-xr-x. 2 root root 6 Apr 29 10:41 Cloud

drwxr-xr-x. 2 root root 6 Apr 29 10:42 DevOps

-rw-r--r--. 1 root root 0 Apr 29 11:00 java17

root@jenkins opt]#

mv <-- this is a command and it used for to move file/directory from one location to another location.

syntax 1:

mv path/file/directory source-path

example:

[root@jenkins var]# cd /opt/

[root@jenkins opt]# ll

total 0

[root@jenkins opt]#

[root@jenkins opt]# ll

total 0

[root@jenkins opt]# pwd

/opt

[root@jenkins opt]# ll /root/cloud/

total 0

-rw-r--r--. 1 root root 0 Apr 29 11:13 a

-rw-r--r--. 1 root root 0 Apr 29 11:30 ab

-rw-r--r--. 1 root root 0 Apr 29 11:30 ac

[root@jenkins opt]# mv /root/cloud/a . <-- pull method.

[root@jenkins opt]# ll

total 0

-rw-r--r--. 1 root root 0 Apr 29 11:13 a

[root@jenkins opt]#

~~~~~~~~~~~~~~~~~~~~~~

syntax: 2:

mv file-nme source-path

push method:

example :

cd /root/cloud
~~~~~~~~~~~~~~~~~~~~~~

[root@jenkins opt]# cd /root/cloud/

[root@jenkins cloud]# ll

total 0

-rw-r--r--. 1 root root 0 Apr 29 11:30 ab

-rw-r--r--. 1 root root 0 Apr 29 11:30 ac

-rw-r--r--. 1 root root 0 Apr 29 11:30 ad

[root@jenkins cloud]# mv ab /opt/

[root@jenkins cloud]# ll /opt/

total 0

-rw-r--r--. 1 root root 0 Apr 29 11:13 a

-rw-r--r--. 1 root root 0 Apr 29 11:30 ab

[root@jenkins cloud]# ll

total 0

-rw-r--r--. 1 root root 0 Apr 29 11:30 ac

-rw-r--r--. 1 root root 0 Apr 29 11:30 ad

-rw-r--r--. 1 root root 0 Apr 29 11:30 ae

[root@jenkins cloud]#

~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~

syntax: 3

mv <source-path> <destination-path>

for example:

cd /var

mv /opt/file-name or directory-name source-path

^^

cp command:

this command used for to copy file/files/directory/Directories to one=path to another-path or to duplicate with new-names on same-data in same-path.

cd - <-- This command used for to move previous path

List all/files/Directories from out side present working directory

syntax:

ls <target-path>

ls -l <target-path>

ls -a <target-path>

ls -la <target-path>

ls -r <target-path>

ls -lr <target-path>

ls -lar <target-path>

by ll command:

ll <target-path>

ll -a <target-path>

ll -r <target-path>

ll -ar<target-path>

VI Editor:

Vi editor Having 3 Modes:

1. Escape Mode
2. Insert Mode

3. Extended Escape Mode <-- here we can apply commands

1. by Default vi enter into Escape mode.

by this Escape mode we can come out from vi editor

TO Enter into the VI Editor for Write/Replace/Edit/Modify/Udate/Delete the Content/Text/Data in file we should enter into Insert mode.

If We want to enter into Insert Mode..!

we have to apply below keys.

i   This key having 2 Attributes out of these <--1 Common Behave that is instert Mode

I   This key having 2 Attributes out of these <--1 Common Behave that is instert Mode

a   This key having 2 Attributes out of these <--1 Common Behave that is instert Mode

A   This key having 2 Attributes out of these <--1 Common Behave that is instert Mode

o   This key having 2 Attributes out of these <--1 Common Behave that is instert Mode

O   This key having 2 Attributes out of these <--1 Common Behave that is instert Mode

~~~~~~

3. Extended Escape Mode <-- here we can apply commands

for example:

to come out from the vi editor
~~~~~~

apply/use keys shift key and hold unit pressing : key

then

:q <-- to come out without saving the document.

:w <-- to save and continue in the document. without coming out from the document

:wq <-- to save and come out from vi editor

:q! <-- to come out forcefully

:w! <-- to save and continue in the document by forcefully.

:wq! <-- to save and come out forcefully

:!appy regular commands

for example:

hostname

cat /etc/os-release

.....

any commands

once you/we apply the command/commands it will execute the output then it will ask you/us press any key to continue.

---

command to know the present using runlevel

runlevel

RUN Levels:

0 <-- init 0 <--this command used for to shutdown the server/machine

1 <-- init 1 <-- this command used for to login into single user mode to mange the system.

2 <-- init 2 <-- this command used for to login into runlevel 2 for working with system with multi user and

without netowrk.

3 <-- init 3 <-- this command used fro to login into runlevel 3 for working with system with multi user and with netowrk.

4 <-- init 4 <-- in our case , we not using this runlevel, this runlevel used by os developers or R&D Team.

5 <-- init 5 <-- this runlevel used for to with GUI

6 <-- init 6 <-- this runlevel used for to restart the server/machine/OS

unsued runlevels:

runlevel -1

runlevel -2

runlevel -4

~~~~~~

whoami

who am i

who

w

~~~~~~

File Permissions:

rwx =readwriteexecute

r=4

w=2

x=1

+ Meand add/give the Permissions

- means denay the Permissions

u=user

g=group

o=others

syntax for giving the Permissions:

file/directory

chmod user/group/other+r/w/x

for example:

to give full Permissions to user:

chmod u+rwx file-nme

to give full Permissions to group

chmod g+rwx file-nme

to give full Permissions to others:

chmod o+rwx file-nme

to give full Permissions to all (user group and others)

chmod ugo+rwx file-nme

---

to Denay the Permissions:

for example:

to denay the Permissions to user:

chmod u-rwx file-nme

to denay the Permissions to group

chmod g-rwx file-nme

to denay the Permissions to others:

chmod o-rwx file-nme

to denay the Permissions to all (user group and others)

chmod ugo-rwx file-nme

~~~~~~

Viewing user information (whoami, id, who, w).

----

whoami:

Syntax: whoami

Description: Displays the username of the current user.

Example:

$ whoami

cloudgen

----

id:

Syntax: id [USERNAME]

Description: Prints real and effective user and group IDs.

Example:

$ id cloudgen

uid=1001(cloudgen) gid=1001(cloudgen) groups=1001(cloudgen),27(sudo)
~~~~~~

---

who:

Syntax: who

Description: Displays information about users who are currently logged in.

Example:

$ who

cloudgen pts/0   2024-05-07 10:00 (192.168.1.100)

---

w:

Syntax: w [USER]

Description: Displays information about the users currently logged in and their processes.

Example:

$ w cloudgen

10:00:12 up 10 days, 1:03, 1 user, load average: 0.02, 0.05, 0.07

USER   TTY   FROM   LOGIN@ IDLE   JCPU   PCPU WHAT

cloudgen pts/0   192.168.1.100   10:00   1:23m 0.02s 0.02s -bash

~~~~~~

Working with USer:

~~~~~

Creating a User: useradd

Explanation: This command is used to create a new user account on the system.

Syntax: sudo useradd [OPTIONS] USERNAME

Command: sudo useradd -m -s /bin/bash cloudgen

Example:

$ sudo useradd -m -s /bin/bash cloudgen

Deleting a User: userdel

Explanation: This command is used to delete a user account from the system.

Syntax: sudo userdel [OPTIONS] USERNAME

Command: sudo userdel -r cloudgen

Example:

$ sudo userdel -r cloudgen

Creating a Group: groupadd

Explanation: This command is used to create a new group on the system.

Syntax: sudo groupadd [OPTIONS] GROUPNAME

Command: sudo groupadd cloudgroup

Example:

$ sudo groupadd cloudgroup

Deleting a Group: groupdel

Explanation: This command is used to delete a group from the system.

Syntax: sudo groupdel [OPTIONS] GROUPNAME

Command: sudo groupdel cloudgroup

Example:

$ sudo groupdel cloudgroup

Changing User Password: passwd

Explanation: This command is used to change a users password.

Syntax: passwd [OPTIONS] USERNAME

Command: passwd cloudgen

Example:

$ passwd cloudgen

Changing Passwords in Bulk: chpasswd

Explanation: This command reads a list of username and password pairs from standard input and updates passwords accordingly.

Syntax: echo "USERNAME:PASSWORD" | sudo chpasswd

Command: echo "cloudgen:secretpassword" | sudo chpasswd

Example:

$ echo "cloudgen:secretpassword" | sudo chpasswd

Modifying User Account Properties: usermod

Explanation: This command is used to modify user account properties.

Syntax: sudo usermod [OPTIONS] USERNAME

Command: sudo usermod -aG cloudgroup cloudgen

Example:

$ sudo usermod -aG cloudgroup cloudgen

Options:

-aG: Append the user to the supplementary group(s) specified.

Modifying Group Properties: groupmod

Explanation: This command is used to modify group properties.

Syntax: sudo groupmod [OPTIONS] GROUPNAME

Command: sudo groupmod -n newcloudgroup cloudgroup

Example:

$ sudo groupmod -n newcloudgroup cloudgroup

Options:

-n newcloudgroup: Rename the group to newcloudgroup.

Starting a New with Different Primary Group: newgrp

Explanation: This command is used to start a new with a different primary group.

Syntax: newgrp [OPTIONS] [GROUPNAME]

Command: newgrp cloudgroup

Example:

$ newgrp cloudgroup

Locking and Unlocking User Accounts: usermod

Explanation: This command can be used to lock and unlock user accounts, preventing login access.

Syntax: sudo usermod -L USERNAME (lock), sudo usermod -U USERNAME (unlock)

Commands:

$ sudo usermod -L cloudgen

$ sudo usermod -U cloudgen

Examples:

Locking the user account:

$ sudo usermod -L cloudgen

Unlocking the user account:

$ sudo usermod -U cloudgen

Managing User Password Aging Policies: chage

Explanation: This command is used to view and modify user password aging policies such as expiry dates and password change requirements.

Syntax: sudo chage [OPTIONS] USERNAME

Commands:

$ sudo chage -l cloudgen

$ sudo chage -E 2025-01-01 cloudgen

Examples:

Viewing password aging information for a user:

$ sudo chage -l cloudgen

Setting an expiry date for the users password:

mathematica

$ sudo chage -E 2025-01-01 cloudgen

Changing User Identification Information: usermod

Explanation: This command can be used to change user identification information such as username, user ID, and group ID.

Syntax: sudo usermod -l NEW_USERNAME OLD_USERNAME

Command:

$ sudo usermod -l newcloudgen cloudgen

Example:

$ sudo usermod -l newcloudgen cloudgen

Changing Group Identification Information: groupmod

Explanation: This command can be used to change group identification information such as group name and group ID.

Syntax: sudo groupmod -n NEW_GROUPNAME OLD_GROUPNAME

Command:

$ sudo groupmod -n newcloudgroup cloudgroup

Example:

$ sudo groupmod -n newcloudgroup cloudgroup

Setting Default User Group: usermod

Explanation: This command is used to set the default group for a user. Files created by the user will be owned by this group.

Syntax: sudo usermod -g GROUPNAME USERNAME

Command:

$ sudo usermod -g cloudgroup cloudgen

Example:

$ sudo usermod -g cloudgroup cloudgen

Assigning a Primary Group to a User: usermod

Explanation: This command is used to assign a primary group to a user. The primary group is the group that is associated with the user upon login.

Syntax: sudo usermod -g GROUPNAME USERNAME

Command:

$ sudo usermod -g cloudgroup cloudgen

Example:

$ sudo usermod -g cloudgroup cloudgen

~~~~~~

working with passwd command

~~~~~~~~~~~~~~

Changing User Passwords: passwd

Explanation: This command is used to change a users password.

Syntax: passwd [OPTIONS] USERNAME

Command:

$ passwd cloudgen

Example:

$ passwd cloudgen

Explanation: When executed without any options, the passwd command prompts the current user (in this case, "cloudgen") to enter their current password and then prompts for a new password. The new password is then set for the specified user.

Changing Another Users Password (as root):

Syntax: sudo passwd USERNAME

Command:

$ sudo passwd cloudgen

Example:

$ sudo passwd cloudgen

Explanation: When executed with sudo, the passwd command allows the root user to change the password for any user on the system. Replace "cloudgen" with the username of the user whose password needs to be changed.

Forcing a User to Change Their Password on Next Login:

Syntax: sudo passwd --expire USERNAME

Command:

$ sudo passwd --expire cloudgen

Example:

$ sudo passwd --expire cloudgen

Explanation: This command sets the password expiry for the specified user ("cloudgen" in this case), requiring them to change their password the next time they log in.

Setting a Specific Expiration Date for a Users Password:

Syntax: sudo passwd --expireat YYYY-MM-DD USERNAME

Command:

$ sudo passwd --expireat 2025-12-31 cloudgen

Example:

$ sudo passwd --expireat 2025-12-31 cloudgen

Explanation: This command sets the password expiry date for the specified user ("cloudgen" in this case) to the specified date (December 31, 2025 in this example).

Locking a Users Password (Disabling Login Access):

Syntax: sudo passwd --lock USERNAME

Command:

$ sudo passwd --lock cloudgen

Example:

$ sudo passwd --lock cloudgen

Explanation: This command locks the password for the specified user ("cloudgen" in this case), effectively preventing them from logging in.

Unlocking a Users Password (Enabling Login Access):

Syntax: sudo passwd --unlock USERNAME

Command:

$ sudo passwd --unlock cloudgen

Example:

$ sudo passwd --unlock cloudgen

Explanation: This command unlocks the password for the specified user ("cloudgen" in this case), allowing them to log in again.

~~~~~~
~

working with file Permissions:

****

File permissions in Linux are a crucial aspect of system security and access control. They determine who can read, write, and execute files and directories on the system. Understanding and managing file permissions is essential for maintaining the security and integrity of a Linux system.

Here is an introduction to the basic concepts of file permissions in Linux:

Types of Entities:

User (owner): The user who owns the file.

Group: The group associated with the file.

Others: All other users who are not the owner and not in the group.
~~~~~~

Permission Types:

Read (r): Allows a user to read the contents of a file or view the contents of a directory.

Write (w): Allows a user to modify the contents of a file or create, delete, or rename files within a directory.

Execute (x): Allows a user to execute a file as a program or access the contents of a directory and traverse it.

Representation:

Symbolic Mode: Represented by characters (r, w, x) for user (u), group (g), and others (o) respectively. For example, rwx represents read, write, and execute permissions.

Numeric Mode: Represented by a three-digit octal number. Each digit represents the permissions for user, group, and others respectively. For example, 755 represents rwxr-xr-x.

Viewing Permissions:

ls -l: Use the ls command with the -l option to view detailed file information, including permissions, ownership, and modification date.

Changing Permissions:

chmod: Use the chmod command to change file permissions. You can modify permissions by using symbolic or numeric representation.

Special Permissions:

Set User ID (SUID): When set on an executable file, it allows the program to run with the permissions of the file owner rather than the user who executed it.

Set Group ID (SGID): Similar to SUID, but the program runs with the permissions of the files group.

Sticky Bit: When set on a directory, it restricts the deletion of files within the directory to only the file owner, the directory owner, or the root user.

Practical examples:

~~~~~~~~~~~~~~~~~~~~~~~~

Changing File Permissions: chmod

Explanation: chmod is used to change the permissions of a file or directory.

Syntax: chmod [OPTIONS] PERMISSIONS FILE

Practical Examples:

Grant read, write, and execute permissions to the user, and read and execute permissions to the group and others:

$ chmod u+rwx,g+rx,o+rx script.sh

~~~~~~~~~~~~~~~~~~~~~~~~

Set permissions using numeric representation (e.g., 755):

$ chmod 755 script.sh

Remove write permissions for others:

$ chmod o-w script.sh

Viewing File Permissions: ls

Explanation: ls is used to list files and directories, and -l option displays detailed information including permissions.

Syntax: ls -l [FILE]

Practical Example:

$ ls -l script.sh

Changing File Ownership: chown

Explanation: chown is used to change the owner and group of a file or directory.

Syntax: chown [OPTIONS] OWNER[:GROUP] FILE

Practical Examples:

Change the owner of a file to a specific user:

$ sudo chown cloudgen script.sh

Change the owner and group of a file:

$ sudo chown cloudgen:admin script.sh

Changing File Group: chgrp

Explanation: chgrp is used to change the group ownership of a file or directory.

Syntax: chgrp [OPTIONS] GROUP FILE

Example:

$ sudo chgrp admin script.sh

Setting Special Permissions: chmod

Explanation: chmod is also used to set special permissions such as Set User ID (SUID), Set Group ID (SGID), and Sticky Bit.

Syntax:

SUID: chmod u+s FILE

SGID: chmod g+s FILE

Sticky Bit:

Example:

chmod +t DIRECTORY

Set SUID bit on a file:

Example:

$ chmod u+s executable_file

Set SGID bit on a directory:

Example:

$ chmod g+s directory

Set Sticky Bit on a directory:

Example:

$ chmod +t directory

~~~~~~

Changing File Permissions: chmod

Explanation: chmod is used to change the permissions of a file or directory.

Syntax: chmod [OPTIONS] PERMISSIONS FILE

Grant read, write, and execute permissions to the user, and read and execute permissions to the group and others:
~~~~~~

Examples:

$ chmod u+rwx,g+rx,o+rx script.sh

Set permissions using numeric representation (e.g., 755):

Examples:

$ chmod 755 script.sh

Remove write permissions for others:

Examples:

$ chmod o-w script.sh

~~~~~~

Understanding File Ownership (User, Group):

Explanation: Each file in Linux is associated with a user (owner) and a group. The owner is the user who created the file, and the group is the group to which the file belongs.
~~~~~~

Example:

$ ls -l script.sh

Changing File Ownership: chown

Explanation: chown is used to change the owner and group of a file or directory.

Syntax: chown [OPTIONS] OWNER[:GROUP] FILE

Change the owner of a file to a specific user:

Examples:

$ sudo chown cloudgen script.sh

Change the owner and group of a file:

Examples:

$ sudo chown cloudgen:admin script.sh

Changing File Group: chgrp

Explanation: chgrp is used to change the group ownership of a file or directory.

Syntax: chgrp [OPTIONS] GROUP FILE

Example:

$ sudo chgrp admin script.sh