Notes by vishalk17 (t.me/vishalk17)
Devops (06.06.2022)

Linking:

Soft Links vs Hard Links

The `ln` command can be used to create two different kinds of links:

- **Soft links (for files / directories)**
- **Hard links (only for files)**

Soft (Symbolic) Links (for files /directories)

A soft link, sometimes called a symbolic link or *symlink*, points to the location or *path* of the original file. It works like a hyperlink on the internet.

Here are a few important aspects of a soft link:

- If the original file is moved or deleted, the symbolic link won't work.
- A soft link can refer to a file on a different file system.
- Soft links are often used to quickly access a frequently-used file without typing the whole location.
- A soft link is something like a shortcut in Windows.

How to Use the ln Command

> By default, the ln command creates hard links. To create a symbolic link, use the -s (--symbolic) option.

To use the `ln` command, open a terminal window and enter the command with the following format:

```
ln [-sf] [source] [destination]
```

- By default, the `ln` command creates a hard link.
- Use the `-s` option to create a soft (symbolic) link.
- The `-f` option will force the command to overwrite a file that already exists.
- *Source* is the file or directory being linked to.

- *Destination* is the location to save the link – if this is left blank, the symlink is stored in the current working directory.

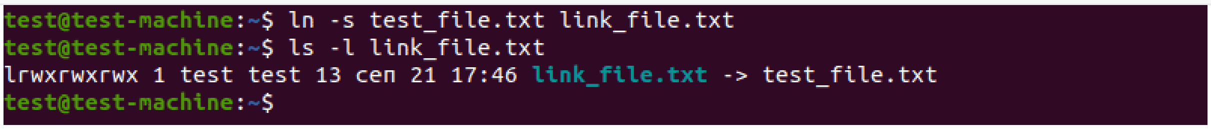
For example, create a symbolic link with:

```
ln -s test_file.txt link_file.txt
```

This creates a symbolic link **(link_file.txt)** that points to the **test_file.txt**.

o verify whether the symlink has been created, use the ls command:

```
ls -l link_file.txt
```



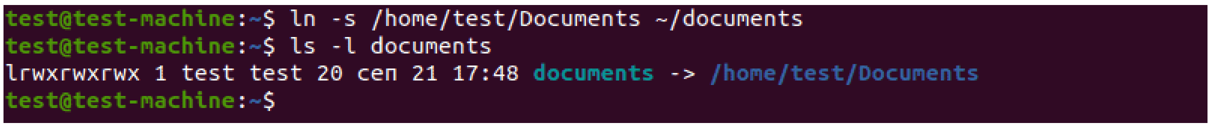
```
test@test-machine:~$ ln -s test_file.txt link_file.txt
test@test-machine:~$ ls -l link_file.txt
lrwxrwxrwx 1 test test 13 сен 21 17:46 link_file.txt -> test_file.txt
test@test-machine:~$
```

Create a Symbolic Link to Linux Directory

A symbolic link can refer to a directory. To create a symbolic link to a directory in Linux:

```
ln -s /mnt/external_drive/stock_photos ~/stock_photos
```

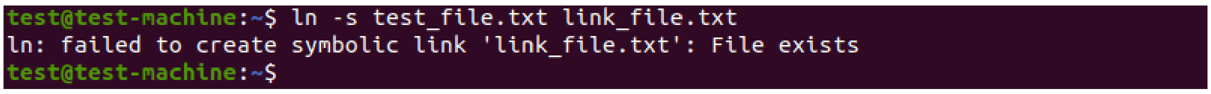
This example creates a symbolic link named **stock_photos** in the **home (~/)** directory. The link refers to the **stock_photos** directory on an **external_drive**.



```
test@test-machine:~$ ln -s /home/test/Documents ~/documents
test@test-machine:~$ ls -l documents
lrwxrwxrwx 1 test test 20 сен 21 17:48 documents -> /home/test/Documents
test@test-machine:~$
```

Force Overwrite Symbolic Links

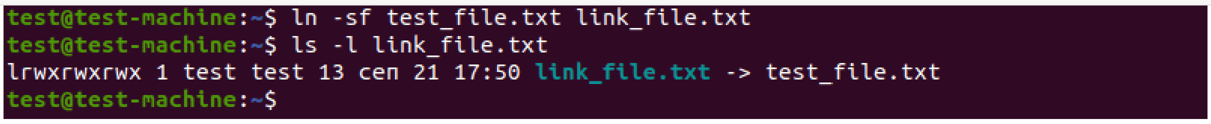
You might receive an error message as displayed in the image below:



```
test@test-machine:~$ ln -s test_file.txt link_file.txt
ln: failed to create symbolic link 'link_file.txt': File exists
test@test-machine:~$
```

The error message means that there's already a file in the destination named **link_file.txt**. Use the `-f` option to force the system to overwrite the destination link:

```
ln -sf test_file.txt link_file.txt
```



```
test@test-machine:~$ ln -sf test_file.txt link_file.txt
test@test-machine:~$ ls -l link_file.txt
lrwxrwxrwx 1 test test 13 сен 21 17:50 link_file.txt -> test_file.txt
test@test-machine:~$
```

Note: Using the `-f` option will permanently delete the existing file.

Hard Links (only for files to link)

When a file is stored on a hard drive, several things happen:

- The data is physically written to the disk.
- A reference file, called *inode*, is created to point to the location of the data.
- A filename is created to refer to the *inode* data.

A hard link works by creating another filename that refers to the *inode* data of the original file. In practice, this is similar to creating a copy of the file.

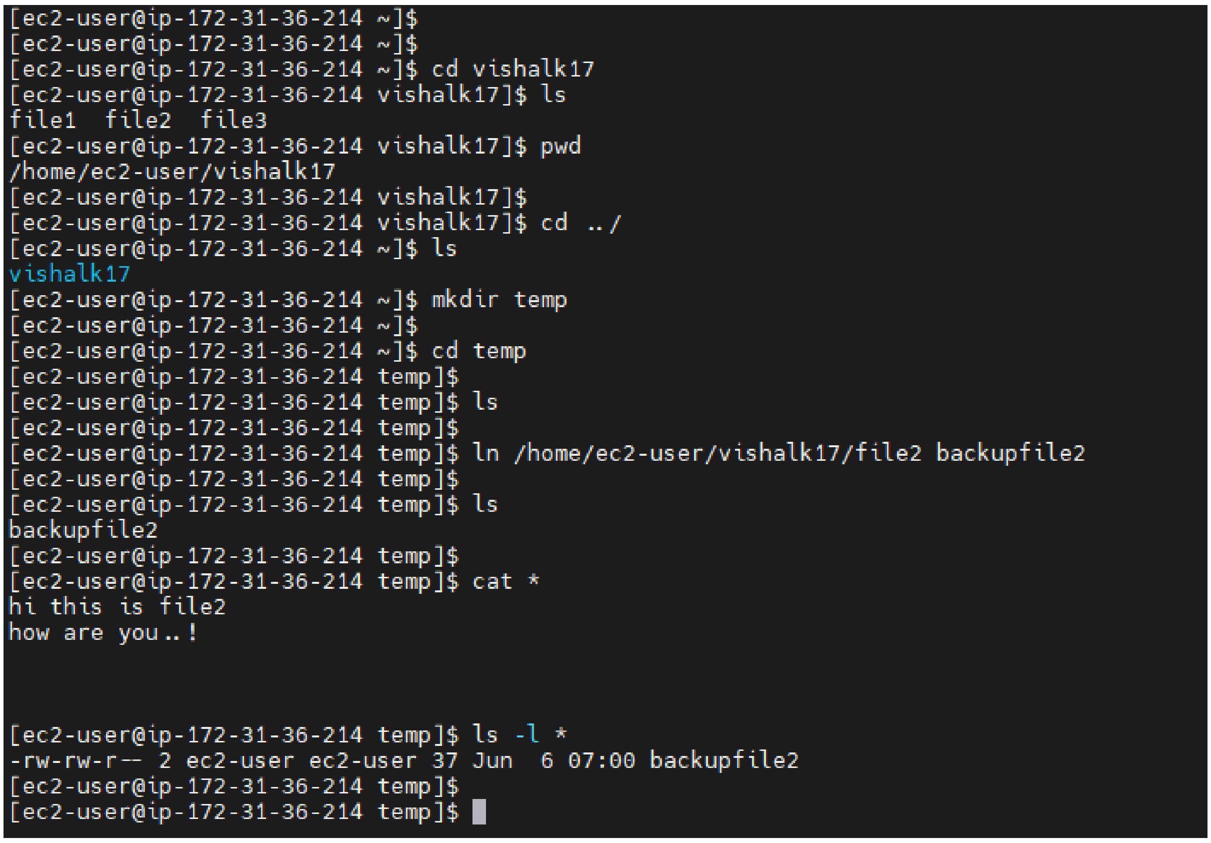
Here are a few important aspects of hard links:

- If the original file is deleted, the file data can still be accessed through other hard links.
- If the original file is moved, hard links still work.
- It just like backup of original file.

To use the `ln` command, open a terminal window and enter the command with the following format:

```
ln [-f] [source] [destination]
```

- By default, the `ln` command creates a hard link.
- The `-f` option will force the command to overwrite a file that already exists.
- *Source* is the file being linked to.
- *Destination* is the location to save the link – if this is left blank, the symlink is stored in the current working directory.

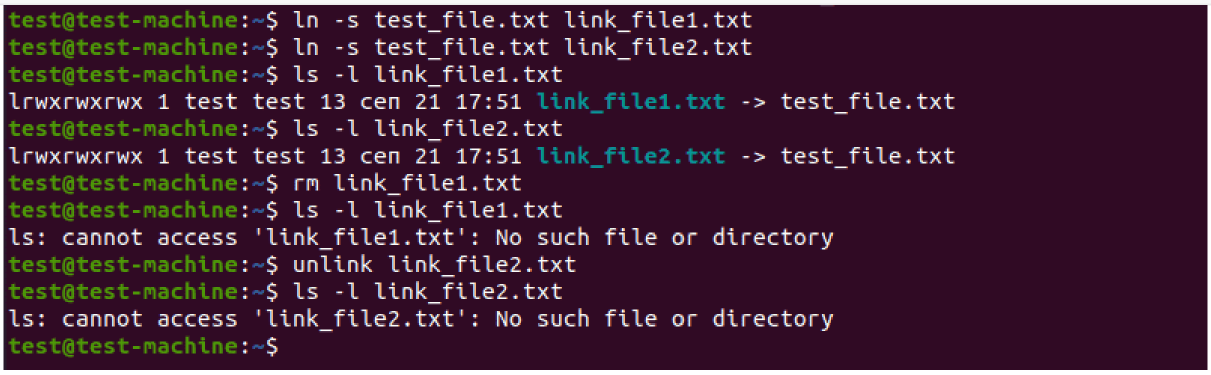


```
[ec2-user@ip-172-31-36-214 ~]$
[ec2-user@ip-172-31-36-214 ~]$
[ec2-user@ip-172-31-36-214 ~]$ cd vishalk17
[ec2-user@ip-172-31-36-214 vishalk17]$ ls
file1  file2  file3
[ec2-user@ip-172-31-36-214 vishalk17]$ pwd
/home/ec2-user/vishalk17
[ec2-user@ip-172-31-36-214 vishalk17]$
[ec2-user@ip-172-31-36-214 vishalk17]$ cd ../
[ec2-user@ip-172-31-36-214 ~]$ ls
vishalk17
[ec2-user@ip-172-31-36-214 ~]$ mkdir temp
[ec2-user@ip-172-31-36-214 ~]$
[ec2-user@ip-172-31-36-214 ~]$ cd temp
[ec2-user@ip-172-31-36-214 temp]$
[ec2-user@ip-172-31-36-214 temp]$ ls
[ec2-user@ip-172-31-36-214 temp]$
[ec2-user@ip-172-31-36-214 temp]$ ln /home/ec2-user/vishalk17/file2 backupfile2
[ec2-user@ip-172-31-36-214 temp]$
[ec2-user@ip-172-31-36-214 temp]$ ls
backupfile2
[ec2-user@ip-172-31-36-214 temp]$
[ec2-user@ip-172-31-36-214 temp]$ cat *
hi this is file2
how are you..!



[ec2-user@ip-172-31-36-214 temp]$ ls -l *
-rw-rw-r-- 2 ec2-user ec2-user 37 Jun  6 07:00 backupfile2
[ec2-user@ip-172-31-36-214 temp]$
[ec2-user@ip-172-31-36-214 temp]$
```

Deleting or Removing Links

If the original file is moved, deleted, or becomes unavailable (such as a server going offline), the link will be unusable. To remove a symbolic link, use either the **`rm`** (remove) or **`unlink`** command:

```
rm link_file.txt
unlink link_file.txt
```

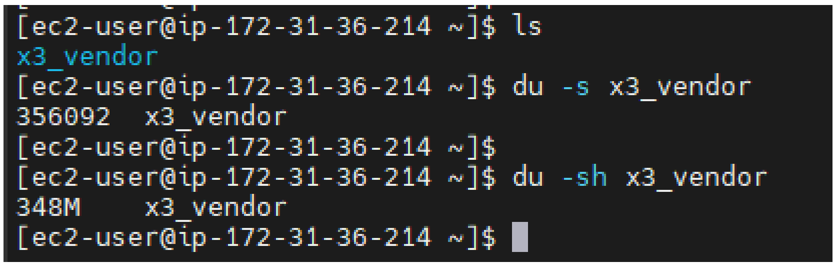


```
test@test-machine:~$ ln -s test_file.txt link_file1.txt
test@test-machine:~$ ln -s test_file.txt link_file2.txt
test@test-machine:~$ ls -l link_file1.txt
lrwxrwxrwx 1 test test 13 сен 21 17:51 link_file1.txt -> test_file.txt
test@test-machine:~$ ls -l link_file2.txt
lrwxrwxrwx 1 test test 13 сен 21 17:51 link_file2.txt -> test_file.txt
test@test-machine:~$ rm link_file1.txt
test@test-machine:~$ ls -l link_file1.txt
ls: cannot access 'link_file1.txt': No such file or directory
test@test-machine:~$ unlink link_file2.txt
test@test-machine:~$ ls -l link_file2.txt
ls: cannot access 'link_file2.txt': No such file or directory
test@test-machine:~$
```

How to Get the Size of a Directory

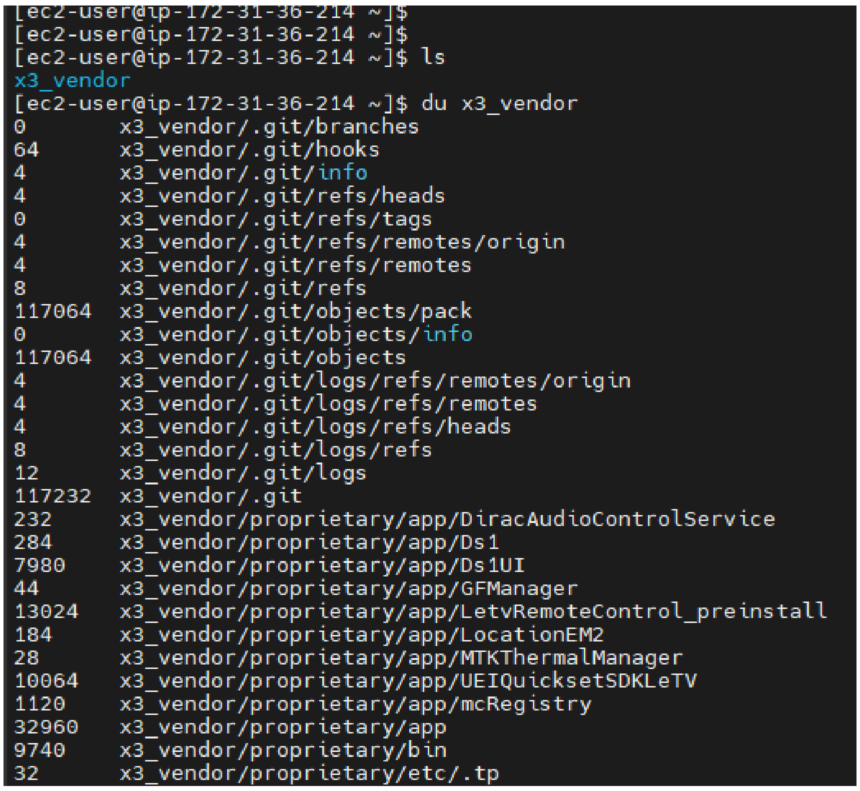
`du` command (short for "disk usage") to get the size

- `-s` : Summarize `du` will only display the total size of the specified directories.
- `-h` : Human readable. `du` will print sizes in human readable format (e.g., 1K, 150M, 2G).



```
[ec2-user@ip-172-31-36-214 ~]$ ls
x3_vendor
[ec2-user@ip-172-31-36-214 ~]$ du -s x3_vendor
356092  x3_vendor
[ec2-user@ip-172-31-36-214 ~]$
[ec2-user@ip-172-31-36-214 ~]$ du -sh x3_vendor
348M    x3_vendor
[ec2-user@ip-172-31-36-214 ~]$
```

Without `-s` option `du` will not only display the size of the specified directory, but also the size of the subdirectories inside of that directory separately.

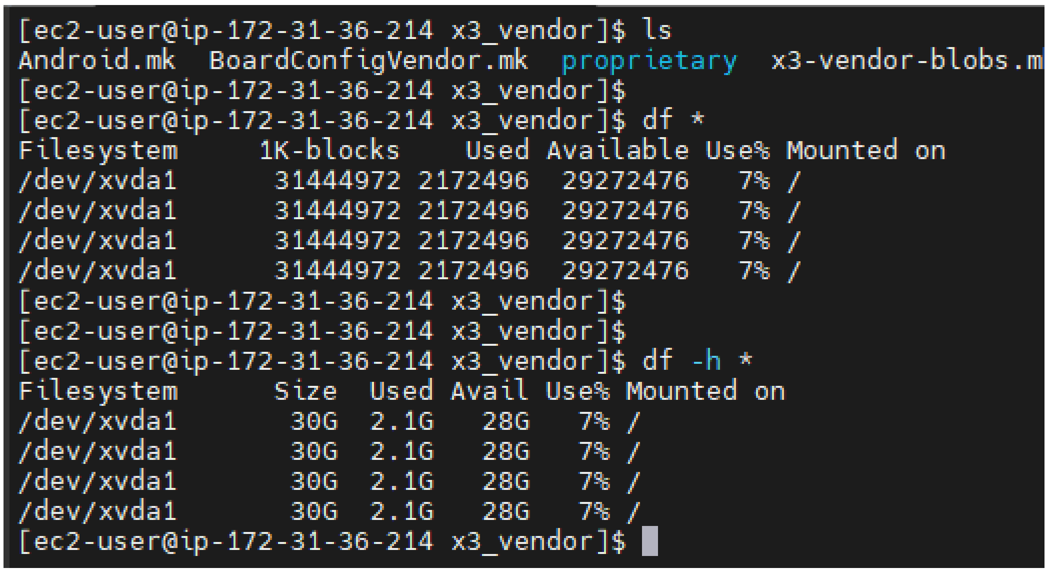


```
[ec2-user@ip-172-31-36-214 ~]$
[ec2-user@ip-172-31-36-214 ~]$
[ec2-user@ip-172-31-36-214 ~]$ ls
x3_vendor
[ec2-user@ip-172-31-36-214 ~]$ du x3_vendor
0       x3_vendor/.git/branches
64      x3_vendor/.git/hooks
4       x3_vendor/.git/info
4       x3_vendor/.git/refs/heads
0       x3_vendor/.git/refs/tags
4       x3_vendor/.git/refs/remotes/origin
4       x3_vendor/.git/refs/remotes
8       x3_vendor/.git/refs
117064  x3_vendor/.git/objects/pack
0       x3_vendor/.git/objects/info
117064  x3_vendor/.git/objects
4       x3_vendor/.git/logs/refs/remotes/origin
4       x3_vendor/.git/logs/refs/remotes
4       x3_vendor/.git/logs/refs/heads
8       x3_vendor/.git/logs/refs
12      x3_vendor/.git/logs
117232  x3_vendor/.git
232     x3_vendor/proprietary/app/DiracAudioControlService
284     x3_vendor/proprietary/app/Ds1
7980    x3_vendor/proprietary/app/Ds1UI
44      x3_vendor/proprietary/app/GFManager
13024   x3_vendor/proprietary/app/LetvRemoteControl_preinstall
184     x3_vendor/proprietary/app/LocationEM2
28      x3_vendor/proprietary/app/MTKThermalManager
10064   x3_vendor/proprietary/app/UEIQuicksetSDKLeTV
1120    x3_vendor/proprietary/app/mcRegistry
32960   x3_vendor/proprietary/app
9740    x3_vendor/proprietary/bin
32      x3_vendor/proprietary/etc/.tp
```

Check Disk Space in Linux

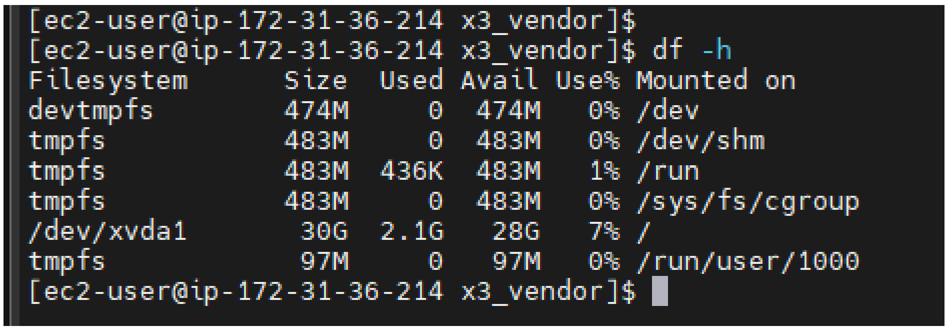
The **df** command (short for disk free), is used to display information related to file systems about total space and available space. by using `'-h'` (prints the results in human-readable format (e.g., **1K 2M 3G**)).

To see the information of the file systems in which currently selected files saved.



```
[ec2-user@ip-172-31-36-214 x3_vendor]$ ls
Android.mk  BoardConfigVendor.mk  proprietary  x3-vendor-blobs.m
[ec2-user@ip-172-31-36-214 x3_vendor]$
[ec2-user@ip-172-31-36-214 x3_vendor]$ df *
Filesystem     1K-blocks    Used Available Use% Mounted on
/dev/xvda1      31444972 2172496  29272476   7% /
/dev/xvda1      31444972 2172496  29272476   7% /
/dev/xvda1      31444972 2172496  29272476   7% /
/dev/xvda1      31444972 2172496  29272476   7% /
[ec2-user@ip-172-31-36-214 x3_vendor]$
[ec2-user@ip-172-31-36-214 x3_vendor]$
[ec2-user@ip-172-31-36-214 x3_vendor]$ df -h *
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1       30G  2.1G   28G   7% /
/dev/xvda1       30G  2.1G   28G   7% /
/dev/xvda1       30G  2.1G   28G   7% /
/dev/xvda1       30G  2.1G   28G   7% /
[ec2-user@ip-172-31-36-214 x3_vendor]$
```

To displays the space available on all currently mounted file systems.



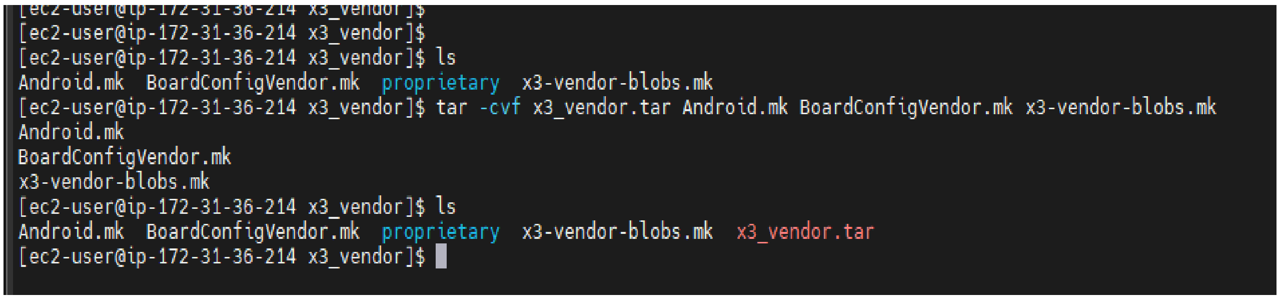
```
[ec2-user@ip-172-31-36-214 x3_vendor]$
[ec2-user@ip-172-31-36-214 x3_vendor]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        474M     0  474M   0% /dev
tmpfs           483M     0  483M   0% /dev/shm
tmpfs           483M  436K  483M   1% /run
tmpfs           483M     0  483M   0% /sys/fs/cgroup
/dev/xvda1       30G  2.1G   28G   7% /
tmpfs            97M     0   97M   0% /run/user/1000
[ec2-user@ip-172-31-36-214 x3_vendor]$
```

Archive

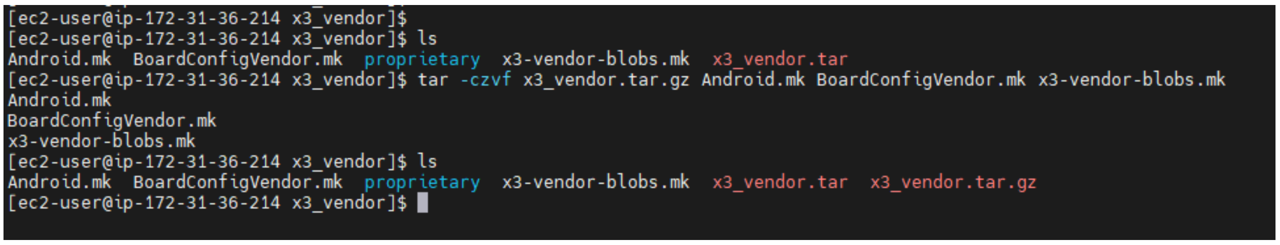
Using tar (Tape Archive) command:

a group of files collected together as one. The term suggests a ball of tar (tarball),

```
archive without gzip compression :

tar -cvf output_file_name.tar file1 file2 dir1 dir2


archive + compresion using gzip :

tar -czvf output_file_name.tar file1 file2 dir1 dir2
```



```
[ec2-user@ip-172-31-36-214 x3_vendor]$
[ec2-user@ip-172-31-36-214 x3_vendor]$ ls
Android.mk  BoardConfigVendor.mk  proprietary  x3-vendor-blobs.mk
[ec2-user@ip-172-31-36-214 x3_vendor]$ tar -cvf x3_vendor.tar Android.mk BoardConfigVendor.mk x3-vendor-blobs.mk
Android.mk
BoardConfigVendor.mk
x3-vendor-blobs.mk
[ec2-user@ip-172-31-36-214 x3_vendor]$ ls
Android.mk  BoardConfigVendor.mk  proprietary  x3-vendor-blobs.mk  x3_vendor.tar
[ec2-user@ip-172-31-36-214 x3_vendor]$
```



```
[ec2-user@ip-172-31-36-214 x3_vendor]$
[ec2-user@ip-172-31-36-214 x3_vendor]$ ls
Android.mk  BoardConfigVendor.mk  proprietary  x3-vendor-blobs.mk  x3_vendor.tar
[ec2-user@ip-172-31-36-214 x3_vendor]$ tar -czvf x3_vendor.tar.gz Android.mk BoardConfigVendor.mk x3-vendor-blobs.mk
Android.mk
BoardConfigVendor.mk
x3-vendor-blobs.mk
[ec2-user@ip-172-31-36-214 x3_vendor]$ ls
Android.mk  BoardConfigVendor.mk  proprietary  x3-vendor-blobs.mk  x3_vendor.tar  x3_vendor.tar.gz
[ec2-user@ip-172-31-36-214 x3_vendor]$
```

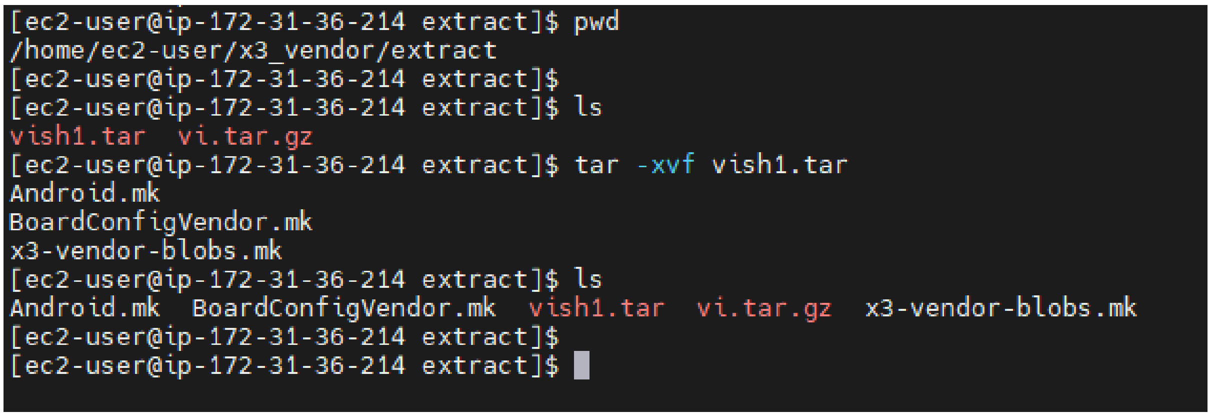
This is what this command is doing:

- -c: Create an archive
- -z: Use gzip to compress the archive
- -v: Enable verbose mode to show the progress of the creation process
- -f: Lets you specify the name of the archive

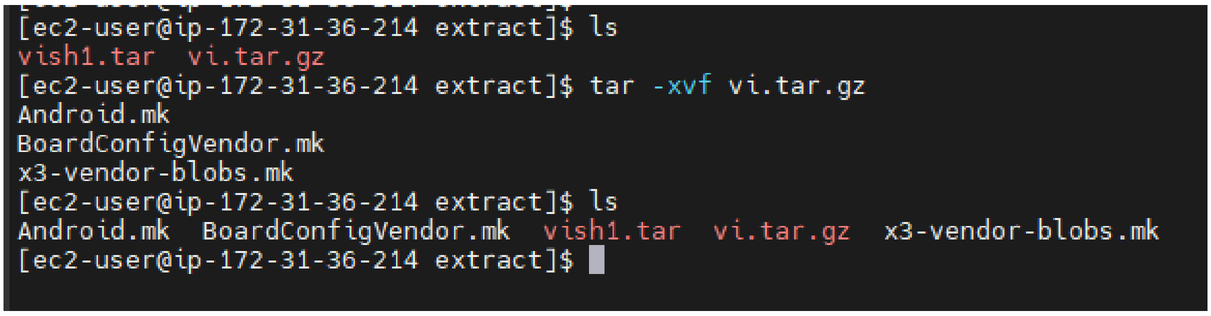
Extract tar archive

The tar command will auto-detect compression type and will extract the archive. The same command can be used to extract tar archives compressed with other algorithms such as **.tar.bz2** .

```
unarchive ( *tar.* ):

tar -xvf input_file_name.tar

or

tar -xvf input_file_name.tar.gz
```

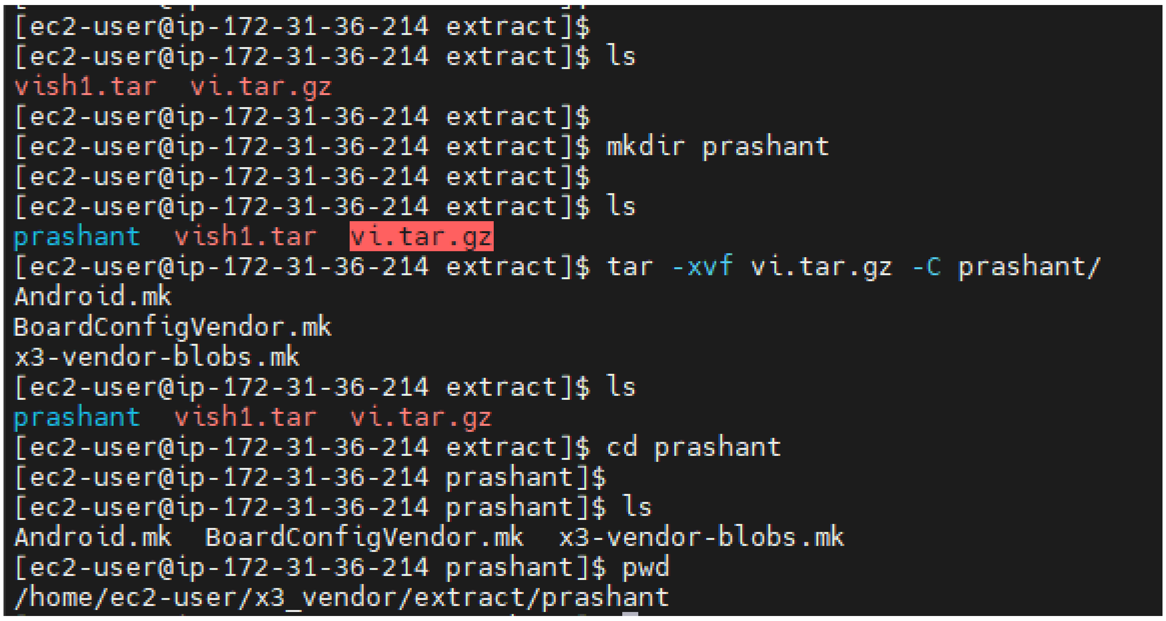


```
[ec2-user@ip-172-31-36-214 extract]$ pwd
/home/ec2-user/x3_vendor/extract
[ec2-user@ip-172-31-36-214 extract]$
[ec2-user@ip-172-31-36-214 extract]$ ls
vish1.tar  vi.tar.gz
[ec2-user@ip-172-31-36-214 extract]$ tar -xvf vish1.tar
Android.mk
BoardConfigVendor.mk
x3-vendor-blobs.mk
[ec2-user@ip-172-31-36-214 extract]$ ls
Android.mk  BoardConfigVendor.mk  vish1.tar  vi.tar.gz  x3-vendor-blobs.mk
[ec2-user@ip-172-31-36-214 extract]$
[ec2-user@ip-172-31-36-214 extract]$
```



```
[ec2-user@ip-172-31-36-214 extract]$ ls
vish1.tar  vi.tar.gz
[ec2-user@ip-172-31-36-214 extract]$ tar -xvf vi.tar.gz
Android.mk
BoardConfigVendor.mk
x3-vendor-blobs.mk
[ec2-user@ip-172-31-36-214 extract]$ ls
Android.mk  BoardConfigVendor.mk  vish1.tar  vi.tar.gz  x3-vendor-blobs.mk
[ec2-user@ip-172-31-36-214 extract]$
```

By default, tar will extract the archive contents in the current working directory . Use the --directory (-C) to extract archive files in a specific directory:



```
[ec2-user@ip-172-31-36-214 extract]$
[ec2-user@ip-172-31-36-214 extract]$ ls
vish1.tar  vi.tar.gz
[ec2-user@ip-172-31-36-214 extract]$
[ec2-user@ip-172-31-36-214 extract]$ mkdir prashant
[ec2-user@ip-172-31-36-214 extract]$
[ec2-user@ip-172-31-36-214 extract]$ ls
prashant  vish1.tar  vi.tar.gz
[ec2-user@ip-172-31-36-214 extract]$ tar -xvf vi.tar.gz -C prashant/
Android.mk
BoardConfigVendor.mk
x3-vendor-blobs.mk
[ec2-user@ip-172-31-36-214 extract]$ ls
prashant  vish1.tar  vi.tar.gz
[ec2-user@ip-172-31-36-214 extract]$ cd prashant
[ec2-user@ip-172-31-36-214 prashant]$
[ec2-user@ip-172-31-36-214 prashant]$ ls
Android.mk  BoardConfigVendor.mk  x3-vendor-blobs.mk
[ec2-user@ip-172-31-36-214 prashant]$ pwd
/home/ec2-user/x3_vendor/extract/prashant
```

gzip :

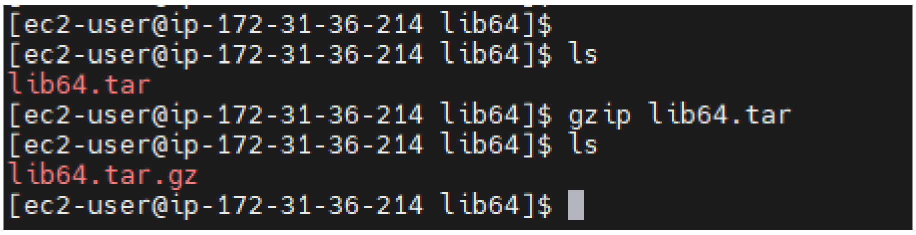
gzip (GNU zip) used for file compression and decompression.

GNU/Linux is a Unix-like operating system made up of different OS components and services that create the Linux OS.

On Linux, gzip is unable to compress a folder, it used to compress a single file only.

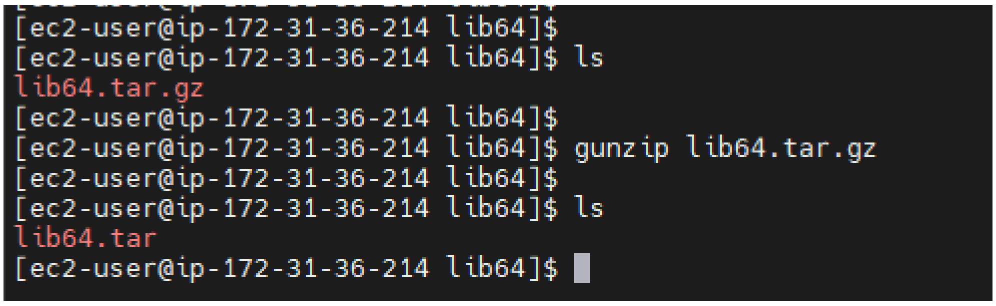
For example : To compress a folder, you should use tar (to archive folder)+ gzip (to compressed tar archive)

```
gzip compression:

gzip output_file_name.tar ( It will compressed tar archive)
```



```
[ec2-user@ip-172-31-36-214 lib64]$
[ec2-user@ip-172-31-36-214 lib64]$ ls
lib64.tar
[ec2-user@ip-172-31-36-214 lib64]$ gzip lib64.tar
[ec2-user@ip-172-31-36-214 lib64]$ ls
lib64.tar.gz
[ec2-user@ip-172-31-36-214 lib64]$
```

```
gzip decompression:

gzip input_file_name.tar.gz ( It will decompressed tar.gz )
```



```
[ec2-user@ip-172-31-36-214 lib64]$
[ec2-user@ip-172-31-36-214 lib64]$ ls
lib64.tar.gz
[ec2-user@ip-172-31-36-214 lib64]$
[ec2-user@ip-172-31-36-214 lib64]$ gunzip lib64.tar.gz
[ec2-user@ip-172-31-36-214 lib64]$
[ec2-user@ip-172-31-36-214 lib64]$ ls
lib64.tar
[ec2-user@ip-172-31-36-214 lib64]$
```

How to Add and Delete Users on linux os

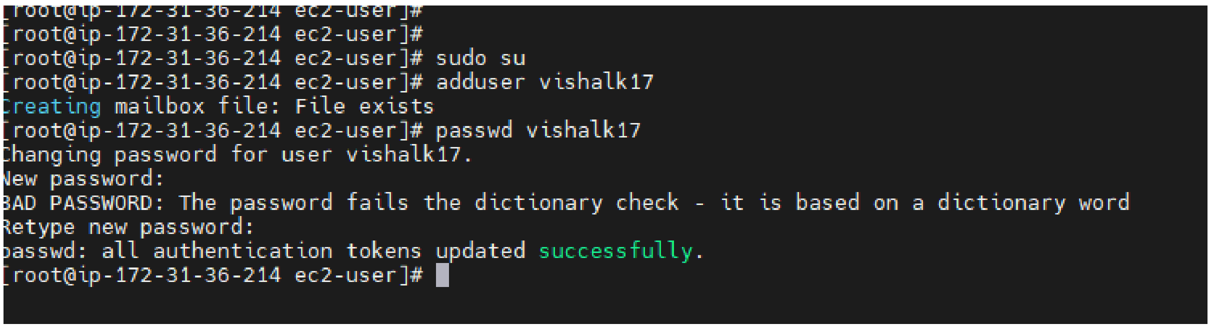
While running as the **root** user gives you complete control over a system and its users, it is also dangerous and possibly destructive. For common system administration tasks, it's a better idea to add an unprivileged user and carry out those tasks without **root** privileges.

For tasks that require administrator privileges, there is a tool installed on Ubuntu systems called sudo. Briefly, sudo allows you to run a command as another user, including users with administrative privileges

Adding a User

If you are signed in as the **root** user, you can create a new user at any time by running the following:

```
sudo su

adduser vishalk17      ( vishalk17 is a newuser to add )
```



```
[root@ip-172-31-36-214 ec2-user]#
[root@ip-172-31-36-214 ec2-user]#
[root@ip-172-31-36-214 ec2-user]# sudo su
[root@ip-172-31-36-214 ec2-user]# adduser vishalk17
Creating mailbox file: File exists
[root@ip-172-31-36-214 ec2-user]# passwd vishalk17
Changing password for user vishalk17.
New password:
BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-36-214 ec2-user]#
```

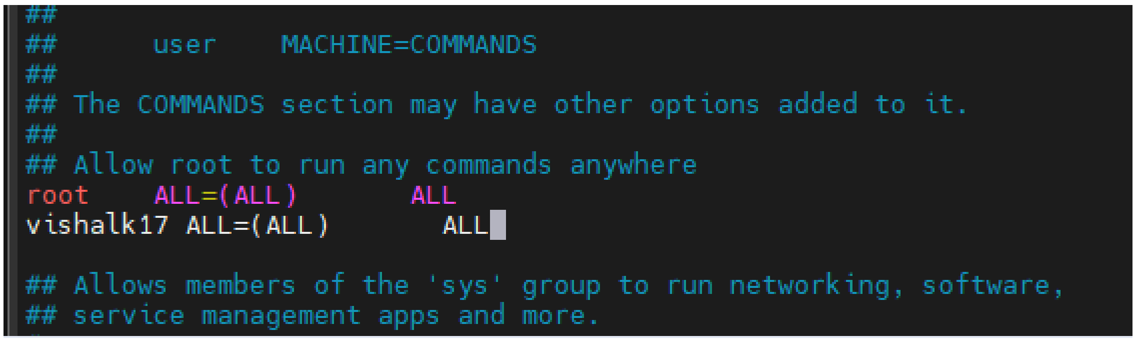
putting your user in the **sudo** group, you can use the visudo command, which opens a configuration file called /etc/sudoers or just vim /etc/sudoers

```
visudo   or  vim /etc/sudoers
```

```
root    ALL=(ALL:ALL) ALL
```

Below this line, add the following line. Be sure to change vishalk17 to the name of the user profile that you would like to grant sudo privileges.

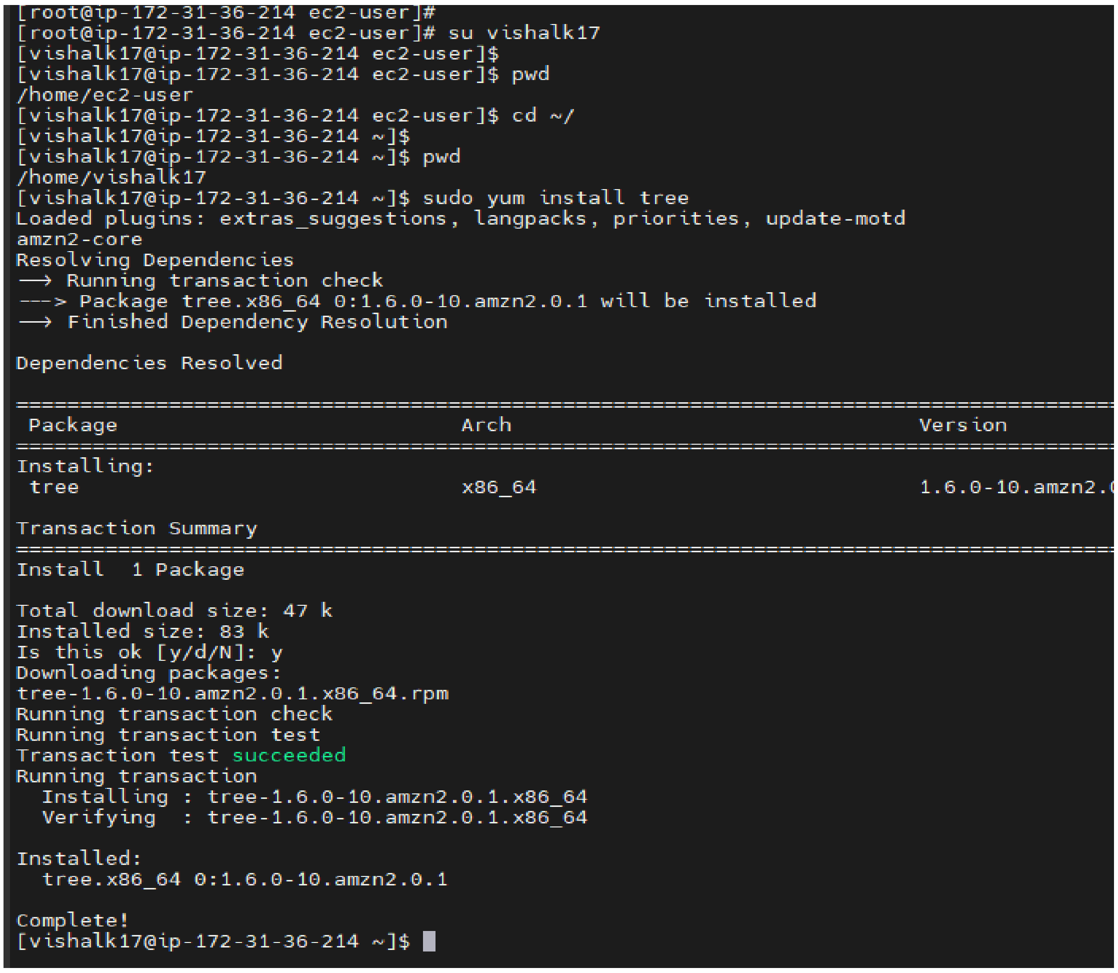
```
root     ALL=(ALL:ALL) ALL
vishalk17     ALL=(ALL:ALL) ALL
```



```
##
##      user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL
vishalk17 ALL=(ALL)       ALL

## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
```

Add a new line like this for each user that should be given full sudo privileges. When you’re finished, save and close the file.

Testing sudo privilege using vishalk17 as a user :

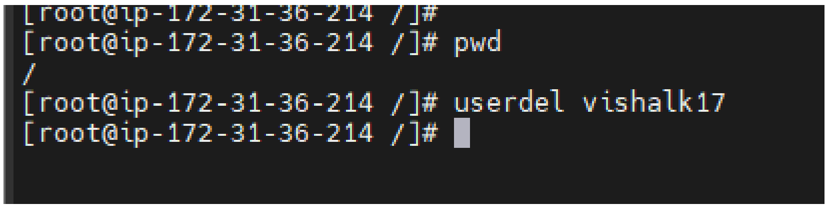


```
[root@ip-172-31-36-214 ec2-user]#
[root@ip-172-31-36-214 ec2-user]# su vishalk17
[vishalk17@ip-172-31-36-214 ec2-user]$
[vishalk17@ip-172-31-36-214 ec2-user]$ pwd
/home/ec2-user
[vishalk17@ip-172-31-36-214 ec2-user]$ cd ~/
[vishalk17@ip-172-31-36-214 ~]$
[vishalk17@ip-172-31-36-214 ~]$ pwd
/home/vishalk17
[vishalk17@ip-172-31-36-214 ~]$ sudo yum install tree
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
---> Package tree.x86_64 0:1.6.0-10.amzn2.0.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====================================================================================
 Package                            Arch                              Version
=====================================================================================
Installing:
 tree                               x86_64                            1.6.0-10.amzn2.

Transaction Summary
=====================================================================================
Install  1 Package

Total download size: 47 k
Installed size: 83 k
Is this ok [y/d/N]: y
Downloading packages:
tree-1.6.0-10.amzn2.0.1.x86_64.rpm
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : tree-1.6.0-10.amzn2.0.1.x86_64
  Verifying  : tree-1.6.0-10.amzn2.0.1.x86_64

Installed:
  tree.x86_64 0:1.6.0-10.amzn2.0.1

Complete!
[vishalk17@ip-172-31-36-214 ~]$
```

Deleting a User

In the event that you no longer need a user, it’s best to delete the old account.

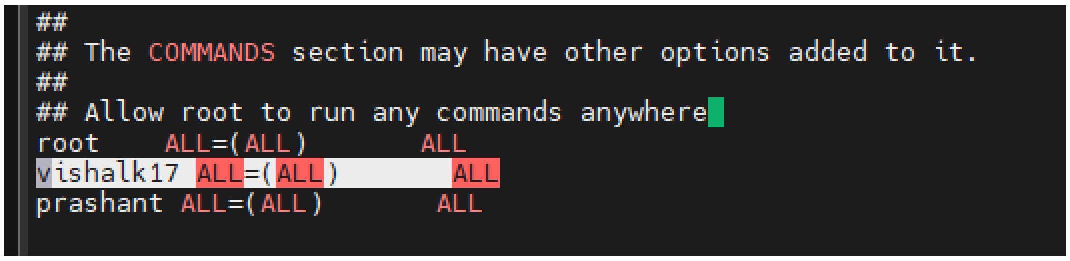
You can delete the user itself, without deleting any of their files, by running the following command as **root**:



```
[root@ip-172-31-36-214 /]#
[root@ip-172-31-36-214 /]# pwd
/
[root@ip-172-31-36-214 /]# userdel vishalk17
[root@ip-172-31-36-214 /]#
```

If you previously configured sudo privileges for the user you deleted, you may want to remove the relevant line again:

```
visudo   or  vim /etc/sudoers
```



```
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root     ALL=(ALL)       ALL
vishalk17 ALL=(ALL)       ALL
prashant ALL=(ALL)       ALL
```

Here I have to delete vishalk17 user line. Because I have deleted user vishalk17.

This will prevent a new user created with the same name from being accidentally given sudo privileges.