LINUX

Introduction

Linux is an operating system that evolved from a kernel created by Linus Torvalds when he was a student at the University of Helsinki.

When Linus Torvalds was studying at the University of Helsinki, he was using a version of the UNIX operating system called 'Minix'. Linus and other users sent requests for modifications and improvements to Minix's creator, Andrew Tanenbaum, but he felt that they weren't necessary. That's when Linus decided to create his own operating system that would take into account users' comments and suggestions for improvements.

Linux Distributions

Following are the Linux distributions.

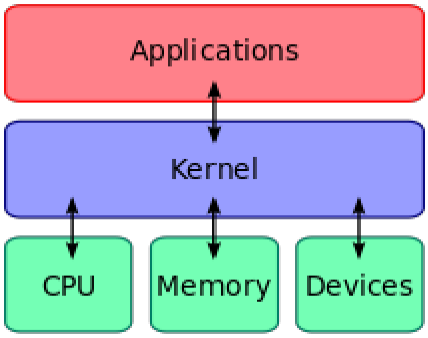
- Redhat Linux
- Ubuntu.
- CentOS
- SuSe Linux
- Debian.
- Fedora
- Gentoo
- Mandriva.
- Slackware

Download different Flavours of Linux distributions

Download Linux from their respective home page. As it is an open source operating system, it's free to download! Not all versions are free, have a look.

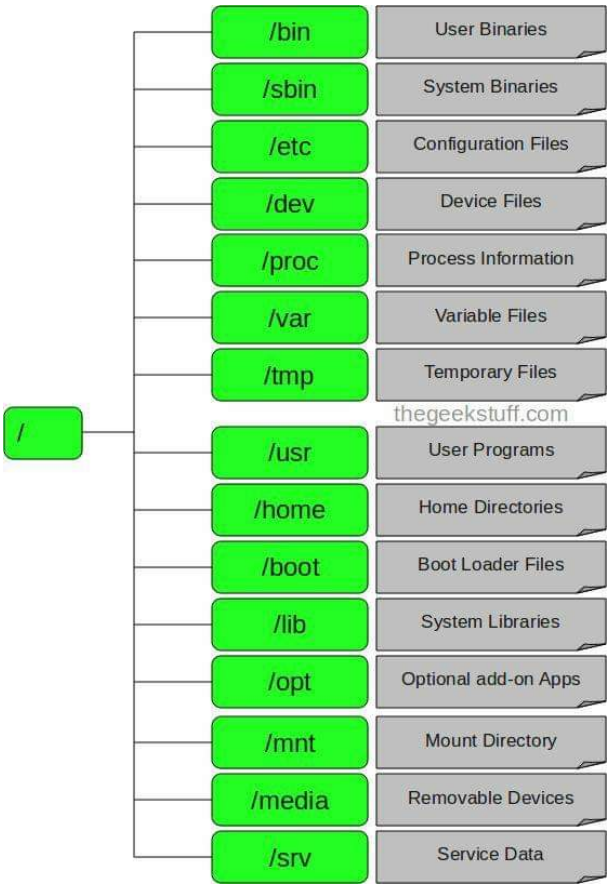
- http://distrocenter.linux.com/
- http://www.redhat.com
- http://fedora.redhat.com/
- http://www.debian.org
- http://www.mandriva.com
- http://www.suse.com
- http://www.opensuse.org/
- http://www.slackware.com
- http://www.ubuntulinux.org/
- http://www.ubuntu.com/
- http://www.distrowatch.com
- http://www.openbsd.org
- http://www.freebsd.org
- http://www.netbsd.org

Kernel Architecture



Kernel is the heart of the operating system. It acts as a bridge between software and hardware. If Software requests the hardware, then kernel delivers the data between software and hardware. For example, if you want to play a song you should launch your default player; it requests the kernel to play a song, now kernel will contact the hardware to seek the permissions or to seek the hardware components like if you plugged in any headset to the device. Most of the Android phones use Linux kernels. We can edit, because it is released under General Public License.

File Directory System



Prepared by PPNREDDY .....

Commands

"mkdir" command

Create the DIRECTORY(ies), if they do not already exist.

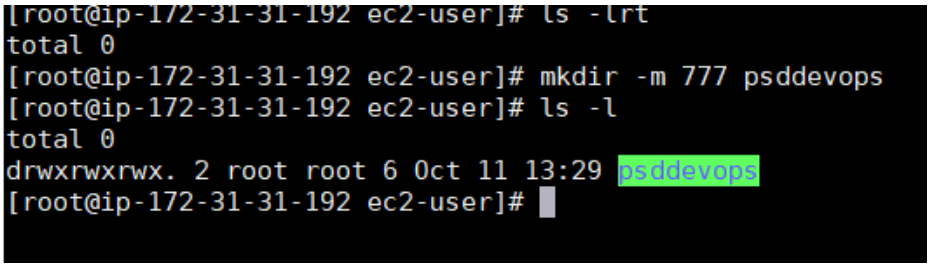
Some options with mkdir are.

-m sets the access mode for the new directory.

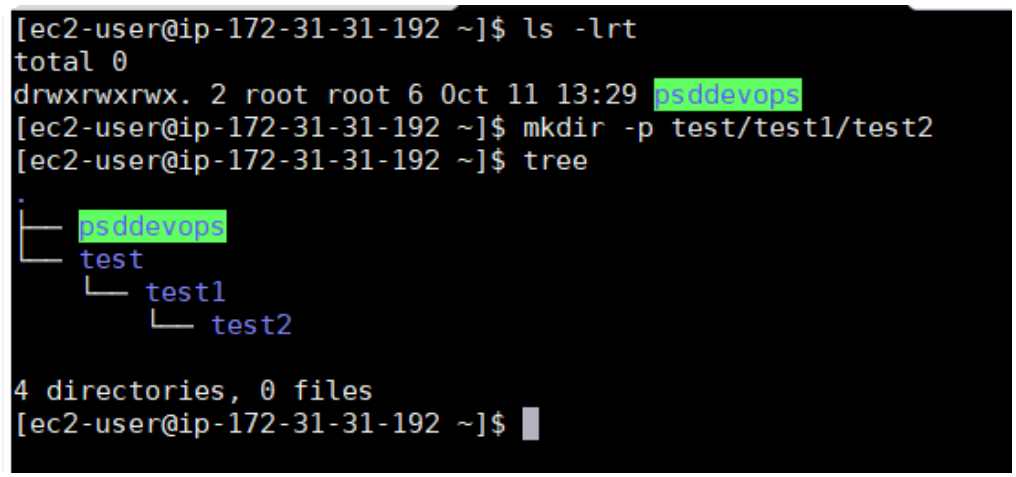
-p if the parent directories don't exist, this command creates them.

-v Print a message for each created directory

1. *mkdir -m 777 psddevops* (It will create the psddevopsfolder with 777 permissions.)



2. *mkdir -p test/test1/test2*



3. *mkdir -v ppreddy*

```
[ec2-user@ip-172-31-31-192 ~]$ mkdir -v ppreddy
mkdir: created directory 'ppreddy'
[ec2-user@ip-172-31-31-192 ~]$ ls -lrt
total 0
drwxrwxrwx. 2 root     root      6 Oct 11 13:29 psddevops
drwxrwxr-x. 3 ec2-user ec2-user 19 Oct 11 13:50 test
drwxrwxr-x. 2 ec2-user ec2-user  6 Oct 11 13:53 ppreddy
[ec2-user@ip-172-31-31-192 ~]$
```

4. *mkdir -v ppreddy{1,2,3,4,5}*

   Note: Same like for *rmdir ppreddy{1,2,3,4,5}*

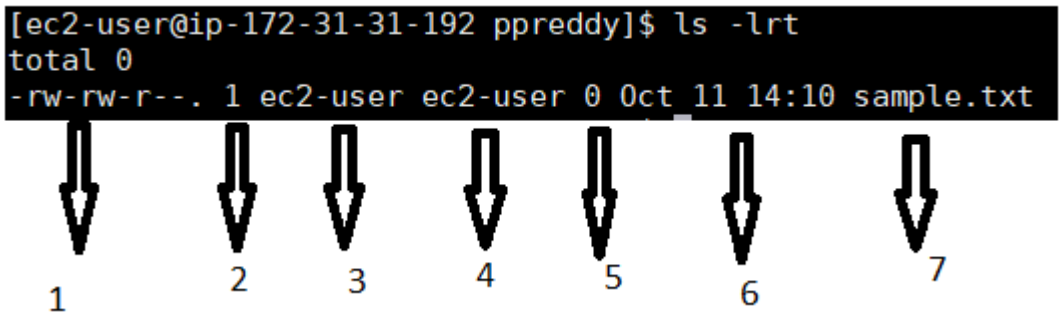
```
[ec2-user@ip-172-31-31-192 ~]$ mkdir -v ppreddy{1,2,3,4,5}
mkdir: created directory 'ppreddy1'
mkdir: created directory 'ppreddy2'
mkdir: created directory 'ppreddy3'
mkdir: created directory 'ppreddy4'
mkdir: created directory 'ppreddy5'
[ec2-user@ip-172-31-31-192 ~]$ ls -lrt
total 0
drwxrwxrwx. 2 root     root      6 Oct 11 13:29 psddevops
drwxrwxr-x. 3 ec2-user ec2-user 19 Oct 11 13:50 test
drwxrwxr-x. 2 ec2-user ec2-user  6 Oct 11 13:53 ppreddy
drwxrwxr-x. 2 ec2-user ec2-user  6 Oct 11 13:56 ppreddy5
drwxrwxr-x. 2 ec2-user ec2-user  6 Oct 11 13:56 ppreddy4
drwxrwxr-x. 2 ec2-user ec2-user  6 Oct 11 13:56 ppreddy3
drwxrwxr-x. 2 ec2-user ec2-user  6 Oct 11 13:56 ppreddy2
drwxrwxr-x. 2 ec2-user ec2-user  6 Oct 11 13:56 ppreddy1
[ec2-user@ip-172-31-31-192 ~]$ rmdir -v ppreddy{1,2,3,4,5}
rmdir: removing directory, 'ppreddy1'
rmdir: removing directory, 'ppreddy2'
rmdir: removing directory, 'ppreddy3'
rmdir: removing directory, 'ppreddy4'
rmdir: removing directory, 'ppreddy5'
[ec2-user@ip-172-31-31-192 ~]$ ls -lrt
total 0
drwxrwxrwx. 2 root     root      6 Oct 11 13:29 psddevops
drwxrwxr-x. 3 ec2-user ec2-user 19 Oct 11 13:50 test
drwxrwxr-x. 2 ec2-user ec2-user  6 Oct 11 13:53 ppreddy
[ec2-user@ip-172-31-31-192 ~]$
```

"ls " command

Lists the contents of your current working directory.

Some options with ls are :

-l list in long format
-a all files including those which begin with dot
-F It is used to classify the directories, executables. It will append indicator (one of */=>@|) to entries
-i list the inode number in the first column
-R recursively list all directories and subdirectories
-r Display in reverse order.
-t sorts files by access time.
-d shows the names of directories and not their contents.
-h Indicates human readable. This mentions file sizes in kilobytes, megabytes, or Gigabytes, instead of just bytes, which is the default setting. Use this optio
With the –l option only.
-S Sorts files by file size. This option is useful only when used together with the option –l



1) Permissions
2) Number of links
3) Owner
4) Group Owner
5) Size of the File in Bytes
6) Date and Time of created
7) File name/Dir Name

```
[ec2-user@ip-172-31-31-192 ppreddy]$ ls -l
total 0
lrwxrwxrwx. 1 ec2-user ec2-user 10 Oct 11 14:20 link_test.txt -> sample.txt
-rw-rw-r--. 1 ec2-user ec2-user  0 Oct 11 14:10 sample.txt
drwxrwxr-x. 3 ec2-user ec2-user 19 Oct 11 14:18 test
[ec2-user@ip-172-31-31-192 ppreddy]$ ls ls
ls: cannot access 'ls': No such file or directory
[ec2-user@ip-172-31-31-192 ppreddy]$ ls
link_test.txt  sample.txt  test
[ec2-user@ip-172-31-31-192 ppreddy]$ ls -a
.  ..  link_test.txt  sample.txt  test
[ec2-user@ip-172-31-31-192 ppreddy]$ ls -r
test  sample.txt  link_test.txt
[ec2-user@ip-172-31-31-192 ppreddy]$ ls -t
link_test.txt  test  sample.txt
[ec2-user@ip-172-31-31-192 ppreddy]$ ls -tl
total 0
lrwxrwxrwx. 1 ec2-user ec2-user 10 Oct 11 14:20 link_test.txt -> sample.txt
drwxrwxr-x. 3 ec2-user ec2-user 19 Oct 11 14:18 test
-rw-rw-r--. 1 ec2-user ec2-user  0 Oct 11 14:10 sample.txt
[ec2-user@ip-172-31-31-192 ppreddy]$ ls -h
link_test.txt  sample.txt  test
[ec2-user@ip-172-31-31-192 ppreddy]$ ls -lh
total 0
lrwxrwxrwx. 1 ec2-user ec2-user 10 Oct 11 14:20 link_test.txt -> sample.txt
-rw-rw-r--. 1 ec2-user ec2-user  0 Oct 11 14:10 sample.txt
drwxrwxr-x. 3 ec2-user ec2-user 19 Oct 11 14:18 test
[ec2-user@ip-172-31-31-192 ppreddy]$ ls -d
.
[ec2-user@ip-172-31-31-192 ppreddy]$ ls -ld
drwxrwxr-x. 3 ec2-user ec2-user 57 Oct 11 14:20 .
[ec2-user@ip-172-31-31-192 ppreddy]$
```

```
[ec2-user@ip-172-31-31-192 ppreddy]$ ls -F
link_test.txt@  sample.txt  test/
```

Here / indicates the directories, * indicates the executable files, @ indicates soft link

```
[ec2-user@ip-172-31-31-192 ppreddy]$ ls -li
total 0
16777375 lrwxrwxrwx. 1 ec2-user ec2-user 10 Oct 11 14:20 link_test.txt -> sample.txt
16777373 -rw-rw-r--. 1 ec2-user ec2-user  0 Oct 11 14:10 sample.txt
13102186 drwxrwxr-x. 3 ec2-user ec2-user 19 Oct 11 14:18 test
[ec2-user@ip-172-31-31-192 ppreddy]$
```

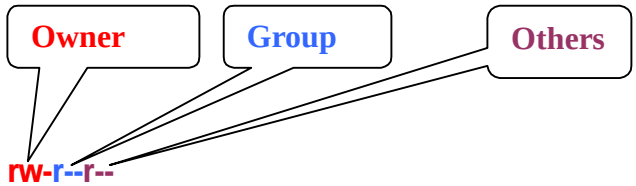
inode is data structure that contain information about files that are created when a file system is created. Each file has an inode and is identified by an inode number in the file system.
When a new file is created, one inode number will assign to that file. The following information is stored in inode.

- Owner and group owner of the file.
- File type
- Permissions on the file.
- Date and time of creation, last read and change.
- Date and time this information has been changed in the inode.
- Number of links to this file
- File size

Default Colour schemes for different file types

Color	File Type
Blue	Directories
White	Text files
Pink	Images
Red	Compressed archives
Cyan	Soft Links
Green	Executables
Yellow	Devices
Flashing red	Broken links

Access Permissions



6 4 4 ----->Octal

r	w	x
4	2	1

r(Read) w(Write) x (Execute) -----> Owner
2 1 0---------- > BinaryNotation
4 + 2 + 0 = 6--------- > Decimal Notation
r(Read) w(Write) x (Execute) ----------> Group
1 0 0---------- > BinaryNotation
r(Read) w(Write) x (Execute) ---------> Others
1 0 0---------- > BinaryNotation

---

cd : To Change working directory.

Syntax : cd << Directory >>
cd <<Path/ Directory>>

Ex:#mkdir /home/ec2-user/ppreddy
cd //home/ec2-user/ppreddy

More about cd command
cd / takes to home directory
cd <<dirname >>
cd << pathname >>

cd ..To move one level up (immediate parent directory)
cd ../.. To move two levels up (so on)
cd ~ It will takes to home directory.
cd It will takes to homedirectory.
cd - It will takes to previous directory.
Relative path cd local
Absoulte path cd /usr/local

```
[ec2-user@ip-172-31-31-192 ~]$ mkdir ppreddy
[ec2-user@ip-172-31-31-192 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-31-192 ~]$ ls -lrt
total 0
drwxrwxr-x. 2 ec2-user ec2-user 6 Oct 13 02:08 ppreddy
[ec2-user@ip-172-31-31-192 ~]$ cd /home/ec2-user/ppreddy/
[ec2-user@ip-172-31-31-192 ppreddy]$ cd ~
[ec2-user@ip-172-31-31-192 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-31-192 ~]$ cd ppreddy/
[ec2-user@ip-172-31-31-192 ppreddy]$ cd
[ec2-user@ip-172-31-31-192 ~]$ cd -
/home/ec2-user/ppreddy
```

What is the difference between absolute path and relative path?
An absolute path begins with the root directory.
Ex: /home/ec2-user/ppreddy/sample.txt
A relative path starts from the currentworking directory. Ex: ./ppreddy/sample.txt

*pwd (Print Working Directory):*To see the current working directory
Ex:
[ec2-user@ip-172-31-31-192 ppreddy]$ pwd
/home/ec2-user/ppreddy

```
[ec2-user@ip-172-31-31-192 ppreddy]$ pwd
/home/ec2-user/ppreddy
[ec2-user@ip-172-31-31-192 ppreddy]$
```

rmdir: Removes a directory.
Some conditions to use rmdir command

- Must be empty before it isdeleted
- Should not be the current directory or a directory at a higher level
- The directory name should exists.
- Should not be HOME directory of the user

Options can be

- -i Interactively asks for confirming the deletion of files. It is useful inavoiding accidental erasure of the file.
- -r Option provides a convenient way to erase a directory even if it is not empty.

- -f Option will forcefully remove a file to which we don't have a write permission.

rm: Removes (unlinks) files or directories.

Syntax: rm <<File/Directory>>

rm -rf ppreddy---------- > Removes directory which has contents.
rm sample.txt----------> Removes the file.
rm *.txt ----------> Removes all the files extension with .txt.

```
[ec2-user@ip-172-31-31-192 ~]$ tree
.
└── ppreddy
    ├── file1.txt
    ├── file2.txt
    ├── file3.txt
    ├── sample.txt
    └── test.csv

1 directory, 5 files
[ec2-user@ip-172-31-31-192 ~]$ cd ppreddy/
[ec2-user@ip-172-31-31-192 ppreddy]$ rm sample.txt
[ec2-user@ip-172-31-31-192 ppreddy]$ ll
total 0
-rw-rw-r--. 1 ec2-user ec2-user 0 Oct 13 02:31 file1.txt
-rw-rw-r--. 1 ec2-user ec2-user 0 Oct 13 02:31 file2.txt
-rw-rw-r--. 1 ec2-user ec2-user 0 Oct 13 02:31 file3.txt
-rw-rw-r--. 1 ec2-user ec2-user 0 Oct 13 02:32 test.csv
[ec2-user@ip-172-31-31-192 ppreddy]$ rm *.txt
[ec2-user@ip-172-31-31-192 ppreddy]$ ll
total 0
-rw-rw-r--. 1 ec2-user ec2-user 0 Oct 13 02:32 test.csv
[ec2-user@ip-172-31-31-192 ppreddy]$ cd ..
[ec2-user@ip-172-31-31-192 ~]$ ll
total 0
drwxrwxr-x. 2 ec2-user ec2-user 22 Oct 13 02:34 ppreddy
[ec2-user@ip-172-31-31-192 ~]$ rm -rf ppreddy
[ec2-user@ip-172-31-31-192 ~]$ ll
total 0
[ec2-user@ip-172-31-31-192 ~]$
```

What is the difference between rm and rmdir?

rm removes the files and directories.

rmdir command removes the directories from the file system. The directory must be empty before it can be remove.

touch : Creates a new file with zero size or update the timestamp of an existing file.

#touch sample.txt

#touch file1.txt file2.txt

```
[ec2-user@ip-172-31-31-192 ~]$ touch sample.txt
[ec2-user@ip-172-31-31-192 ~]$ touch file1.txt file2.txt
[ec2-user@ip-172-31-31-192 ~]$ ls -lrt
total 0
drwxrwxr-x. 2 ec2-user ec2-user 6 Oct 13 02:50 ppreddy
-rw-rw-r--. 1 ec2-user ec2-user 0 Oct 13 02:51 sample.txt
-rw-rw-r--. 1 ec2-user ec2-user 0 Oct 13 02:51 file2.txt
-rw-rw-r--. 1 ec2-user ec2-user 0 Oct 13 02:51 file1.txt
```

We can use below command for creating a new file.

#cat > newcatfile.txt (type your text when done press Ctrl + c).

```
[ec2-user@ip-172-31-31-192 ~]$ cat > devops.txt
Hi This is sample devops file
^C
[ec2-user@ip-172-31-31-192 ~]$ ll
total 4
-rw-rw-r--. 1 ec2-user ec2-user 30 Oct 13 02:56 devops.txt
-rw-rw-r--. 1 ec2-user ec2-user  0 Oct 13 02:51 file1.txt
-rw-rw-r--. 1 ec2-user ec2-user  0 Oct 13 02:51 file2.txt
drwxrwxr-x. 2 ec2-user ec2-user  6 Oct 13 02:50 ppreddy
-rw-rw-r--. 1 ec2-user ec2-user  0 Oct 13 02:51 sample.txt
[ec2-user@ip-172-31-31-192 ~]$ cat devops.txt
Hi This is sample devops file
```

Note: If data is already exists in that file, touch command will not overwrite, just it will update the timestamp.

```
[ec2-user@ip-172-31-31-192 ~]$ ls -lrt
total 4
drwxrwxr-x. 2 ec2-user ec2-user  6 Oct 13 02:50 ppreddy
-rw-rw-r--. 1 ec2-user ec2-user  0 Oct 13 02:51 sample.txt
-rw-rw-r--. 1 ec2-user ec2-user  0 Oct 13 02:51 file2.txt
-rw-rw-r--. 1 ec2-user ec2-user  0 Oct 13 02:51 file1.txt
-rw-rw-r--. 1 ec2-user ec2-user 30 Oct 13 02:56 devops.txt
[ec2-user@ip-172-31-31-192 ~]$ touch devops.txt
[ec2-user@ip-172-31-31-192 ~]$ ls -lrt
total 4
drwxrwxr-x. 2 ec2-user ec2-user  6 Oct 13 02:50 ppreddy
-rw-rw-r--. 1 ec2-user ec2-user  0 Oct 13 02:51 sample.txt
-rw-rw-r--. 1 ec2-user ec2-user  0 Oct 13 02:51 file2.txt
-rw-rw-r--. 1 ec2-user ec2-user  0 Oct 13 02:51 file1.txt
-rw-rw-r--. 1 ec2-user ec2-user 30 Oct 13 02:59 devops.txt
[ec2-user@ip-172-31-31-192 ~]$
```

locate:

Using locate command you can quickly search for the location of a specific file (or group of files

#locate mithun.txt

The example below shows all files in the system that contains the word crontab in it.

#locate Crontab

```
[ec2-user@ip-172-31-31-192 ppreddy]$ locate sample.txt
/home/ec2-user/sample.txt
[ec2-user@ip-172-31-31-192 ppreddy]$ locate -i sample.txt
/home/ec2-user/sample.txt
[ec2-user@ip-172-31-31-192 ppreddy]$ locate crontab
/etc/anacrontab
/etc/crontab
/usr/bin/crontab
/usr/share/licenses/crontabs
/usr/share/licenses/crontabs/COPYING
/usr/share/man/man1/crontab.1.gz
/usr/share/man/man4/crontabs.4.gz
/usr/share/man/man5/anacrontab.5.gz
/usr/share/man/man5/crontab.5.gz
[ec2-user@ip-172-31-31-192 ppreddy]$
```

```
[ec2-user@ip-172-31-31-192 ppreddy]$ touch myfile.txt
[ec2-user@ip-172-31-31-192 ppreddy]$ locate myfile.txt
[ec2-user@ip-172-31-31-192 ppreddy]$ sudo updatedb
[ec2-user@ip-172-31-31-192 ppreddy]$ locate myfile.txt
/home/ec2-user/ppreddy/myfile.txt
[ec2-user@ip-172-31-31-192 ppreddy]$
```

#updatedb : Only root user can able to run.

find:

find command used to search and locate list of files and directories based on conditions you specify for files that match the arguments. Find can be used in variety of conditions like you can find files by permissions, users, groups, file type, date, size and other possible criteria.

#find ~ -name psddevops.sh
It will search the psddevops.sh file under logged in user home directory.

#find . -name psddevops.sh
It will search the psddevops.sh file under current directory.

#find . -iname psddevops.sh
It will search the psddevops.sh file under current directory ignoring the case.

find . -type f -perm 0777: It will search all the files which have 777 permissions.

```
[ec2-user@ip-172-31-31-192 ppreddy]$ cd
[ec2-user@ip-172-31-31-192 ~]$ find ~ -name psddevops.sh
/home/ec2-user/ppreddy/psddevops.sh
[ec2-user@ip-172-31-31-192 ~]$ cd /home/ec2-user/ppreddy
[ec2-user@ip-172-31-31-192 ppreddy]$ find . -name psddevops.sh
./psddevops.sh
[ec2-user@ip-172-31-31-192 ppreddy]$ find . -iname psddevops.sh
./psddevops.sh
./PsdDevops.sh
[ec2-user@ip-172-31-31-192 ppreddy]$ find . -type f -perm 0777
./PsdDevops.sh
[ec2-user@ip-172-31-31-192 ppreddy]$
```

find . -perm /a=x: It will search all the files which have execute permissions.
find . -type f -empty: It will search all the empty files.

```
[ec2-user@ip-172-31-31-192 ppreddy]$ ll
total 0
-rw-rw-r--. 1 ec2-user ec2-user 0 Oct 13 03:10 myfile.txt
-rw-rw-r--. 1 ec2-user ec2-user 0 Oct 13 05:25 psddevops.sh
-rwxrwxrwx. 1 ec2-user ec2-user 0 Oct 13 05:28 PsdDevops.sh
-rw-rw-r--. 1 ec2-user ec2-user 0 Oct 13 05:34 test.sh
[ec2-user@ip-172-31-31-192 ppreddy]$ find . -perm /a=x
.
./PsdDevops.sh
[ec2-user@ip-172-31-31-192 ppreddy]$ find . -type f -empty
./myfile.txt
./psddevops.sh
./PsdDevops.sh
./test.sh
[ec2-user@ip-172-31-31-192 ppreddy]$ cat > test.sh
Hi^C
[ec2-user@ip-172-31-31-192 ppreddy]$ find . -type f -empty
./myfile.txt
./psddevops.sh
./PsdDevops.sh
./test.sh
[ec2-user@ip-172-31-31-192 ppreddy]$ vi test.sh
[ec2-user@ip-172-31-31-192 ppreddy]$ find . -type f -empty
./myfile.txt
./psddevops.sh
./PsdDevops.sh
[ec2-user@ip-172-31-31-192 ppreddy]$
```

find . -type f –name ".*": It will search all the hidden files in current directory.

```
[ec2-user@ip-172-31-31-192 ppreddy]$ find . -type f -name ".*"
./.mytext.csv
```

find / -name <<FIleName>> : It will display the all files which we provide the name.

ex.# find / -name psddevops.sh: It will display the all the locations, where the file psddevops.shis available.

#find / -iname psddevops.sh: It will display the all the locations, where the file psddevops.shis available.Here case insensitive.

```
[ec2-user@ip-172-31-31-192 /]$ sudo find / -iname psddevops.sh
/home/ec2-user/ppreddy/psddevops.sh
/home/ec2-user/ppreddy/PsdDevops.sh
[ec2-user@ip-172-31-31-192 /]$ sudo find / -name psddevops.sh
/home/ec2-user/ppreddy/psddevops.sh
[ec2-user@ip-172-31-31-192 /]$
```

#find . -mtime 1 : It will find all the files modified exact 1 day in current directory.

#find / -mtime 1 : It will find all the files modified exact 1 day in all directories.

#find . -mtime -1 : It will find all the files modified less than 1 day

#find . -mtime +1 : It will find all the files modified more than 1 day

```
[ec2-user@ip-172-31-31-192 ~]$ sudo find . -mtime 1
./.ssh
./.ssh/authorized_keys
[ec2-user@ip-172-31-31-192 ~]$ sudo find . -mtime -1
.
./.bash_history
./ppreddy
./ppreddy/myfile.txt
./ppreddy/psddevops.sh
./ppreddy/PsdDevops.sh
./ppreddy/test.sh
./ppreddy/.mytext.csv
./sample.txt
./file1.txt
./file2.txt
./devops.txt
[ec2-user@ip-172-31-31-192 ~]$ sudo find . -mtime +1
./.bash_logout
./.bash_profile
./.bashrc
[ec2-user@ip-172-31-31-192 ~]$
```

#find / -mmin -10 : It will locate/display the files which modified less than 10 minutes ago.

```
[ec2-user@ip-172-31-31-192 ~]$ sudo find . -mmin -10
[ec2-user@ip-172-31-31-192 ~]$ sudo find . -mmin +10
.
./.bash_logout
./.bash_profile
./.bashrc
./.ssh
./.ssh/authorized_keys
./.bash_history
./ppreddy
./ppreddy/myfile.txt
./ppreddy/psddevops.sh
./ppreddy/PsdDevops.sh
./ppreddy/test.sh
./ppreddy/.mytext.csv
./sample.txt
./file1.txt
./file2.txt
./devops.txt
[ec2-user@ip-172-31-31-192 ~]$ touch hello.txt
[ec2-user@ip-172-31-31-192 ~]$ sudo find . -mmin -10
.
./hello.txt
[ec2-user@ip-172-31-31-192 ~]$
```

What the difference is between locate and find?

The find command has several options and is very configurable. There are many ways to reduce the depth and breadth of your search and make it more efficient.

locate uses a previously built database, If database is not updated then locate command will not show the output. to sync the database it is must to execute updatedb command.

By default, the system will run updatedb which takes a snapshot of the system files once a day, locate uses this snapshot to quickly report what files are where. However, recent file additions or removals (within 24 hours) are not recorded in the snapshot and are unknown to locate.

#updatedb : Only root user can able to run.

umask:

User Mask or User file creation MASK : It is used to set the permissions for files/directories newly created on Linux Machine.

- The default umask 002 used for normal user. With this mask default directory permission are drwxrwxr-x (775) and default file permissions are -rw-rw--r-- (664).
- The default umask for the root user is 022 result into default directory permissions are drwxr-xr-x (755) and default file permissions are -rw-r--r—(644).
- For directories, the base permissions are (rwxrwxrwx) 0777 and for files they are 0666 (rw-rw-rw).
- umask setup is in /etc/bashrc or /etc/profile file as follows.

Below sample code from /etc/bashrc file.

```
# By default, we want this to get set.
# Even for non-interactive, non-login shells.
if [ $UID -gt 99 ] && [ "`id -gn`" = "`id -un`" ]; then
umask 002

else
umask 022
fi
```

Calculating the permissions for Files and directories are when umask value is 006.

For File:

Subtract the umask value from base permissions. 666 – 006 = 660 (rw-rw)

For Directory:

Subtract the umask value from base permissions. 777 – 006 = 771 (rwxrwx--x)

If you want to know the umask value simply type unmask as follows.

#umask

If you want to set the umask values as follows.

#umask 222

```
[ec2-user@ip-172-31-31-192 ~]$ umask
0002
[ec2-user@ip-172-31-31-192 ~]$ sudo su
[root@ip-172-31-31-192 ec2-user]# umask
0022
[root@ip-172-31-31-192 ec2-user]#
```

```
[ec2-user@ip-172-31-31-192 ~]$ mkdir folder1
[ec2-user@ip-172-31-31-192 ~]$ sudo mkdir folder2
[ec2-user@ip-172-31-31-192 ~]$ ls -lrt
total 4
-rw-rw-r--. 1 ec2-user ec2-user  0 Oct 13 02:51 sample.txt
-rw-rw-r--. 1 ec2-user ec2-user  0 Oct 13 02:51 file2.txt
-rw-rw-r--. 1 ec2-user ec2-user  0 Oct 13 02:51 file1.txt
-rw-rw-r--. 1 ec2-user ec2-user 30 Oct 13 02:59 devops.txt
drwxrwxr-x. 2 ec2-user ec2-user 98 Oct 13 05:40 ppreddy
-rw-rw-r--. 1 ec2-user ec2-user  0 Oct 13 05:52 hello.txt
drwxrwxr-x. 2 ec2-user ec2-user  6 Oct 13 06:12 folder1
drwxr-xr-x. 2 root     root      6 Oct 13 06:13 folder2
[ec2-user@ip-172-31-31-192 ~]$
```

chmod : Change file access permissions.

Syntax :

chmod << options >><<access permissions >><< filename/directory >>

777 > Change permission of the file, make it to executable by all (Owner, Group and Others).

400 >Change the permission of the file – only Owner can read.

#chmod -R 777 sample_folder

```
[ec2-user@ip-172-31-31-192 ~]$ chmod -R 777 sample_folder
[ec2-user@ip-172-31-31-192 ~]$ cd sample_folder
[ec2-user@ip-172-31-31-192 sample_folder]$ ll
total 0
-rwxrwxrwx. 1 ec2-user ec2-user 0 Oct 15 02:32 psddevops.txt
[ec2-user@ip-172-31-31-192 sample_folder]$
```

Here R represents recursively.

Following example shows another way of giving the permissions.

chmod u-rwx devops.txt

chmod og-rwx Bachi.txt # ls -l

chmod ugo+rwx devops.txt

chmod ugo-rwx devops.txt

chmod u+rwx,g=r-x,o-x devops.txt

#chmod u+rwx,g=r-x,o-x devops.txt

#chmod u+rwx,g=r+x,o+x devops.txt

+represents adds the designated permission(s) to a file or directory.

-represents removes the designated permission(s) from a file or directory.

= represents Sets the designated permission(s)

```
[ec2-user@ip-172-31-31-192 sample_folder]$ ll
total 0
-rw-rw-r--. 1 ec2-user ec2-user 0 Oct 15 02:37 devops.txt
-rwxrwxrwx. 1 ec2-user ec2-user 0 Oct 15 02:32 psddevops.txt
[ec2-user@ip-172-31-31-192 sample_folder]$ chmod u-rwx devops.txt
[ec2-user@ip-172-31-31-192 sample_folder]$ ll
total 0
----rw-r--. 1 ec2-user ec2-user 0 Oct 15 02:37 devops.txt
-rwxrwxrwx. 1 ec2-user ec2-user 0 Oct 15 02:32 psddevops.txt
[ec2-user@ip-172-31-31-192 sample_folder]$ chmod og-rwx devops.txt
[ec2-user@ip-172-31-31-192 sample_folder]$ ll
total 0
----------. 1 ec2-user ec2-user 0 Oct 15 02:37 devops.txt
-rwxrwxrwx. 1 ec2-user ec2-user 0 Oct 15 02:32 psddevops.txt
[ec2-user@ip-172-31-31-192 sample_folder]$ chmod ugo+rwx devops.txt
[ec2-user@ip-172-31-31-192 sample_folder]$ ll
total 0
-rwxrwxrwx. 1 ec2-user ec2-user 0 Oct 15 02:37 devops.txt
-rwxrwxrwx. 1 ec2-user ec2-user 0 Oct 15 02:32 psddevops.txt
[ec2-user@ip-172-31-31-192 sample_folder]$ chmod ugo-rwx devops.txt
[ec2-user@ip-172-31-31-192 sample_folder]$ ll
total 0
----------. 1 ec2-user ec2-user 0 Oct 15 02:37 devops.txt
-rwxrwxrwx. 1 ec2-user ec2-user 0 Oct 15 02:32 psddevops.txt
[ec2-user@ip-172-31-31-192 sample_folder]$ chmod u+rwx,g=r-x,o-x devops.txt
[ec2-user@ip-172-31-31-192 sample_folder]$ ll
total 0
-rwxr-----. 1 ec2-user ec2-user 0 Oct 15 02:37 devops.txt
-rwxrwxrwx. 1 ec2-user ec2-user 0 Oct 15 02:32 psddevops.txt
[ec2-user@ip-172-31-31-192 sample_folder]$ chmod u+rwx,g=r+x,o+x devops.txt
[ec2-user@ip-172-31-31-192 sample_folder]$ ll
total 0
-rwxr-x--x. 1 ec2-user ec2-user 0 Oct 15 02:37 devops.txt
-rwxrwxrwx. 1 ec2-user ec2-user 0 Oct 15 02:32 psddevops.txt
[ec2-user@ip-172-31-31-192 sample_folder]$
```

#	Octal Permission Representation	Permission Reference
0	No permission	---
1	Execute permission	--x
2	Write permission	-w-
3	Execute and write permission: 1 (execute) + 2 (write) = 3	-wx
4	Read permission	r--
5	Read and execute permission: 4 (read) + 1 (execute) = 5	r-x
6	Read and write permission: 4 (read) + 2 (write) = 6	rw-
7	All permissions: 4 (read) + 2 (write) + 1 (execute) = 7	rwx

chown : Change file owner and group.

chown ansadmin devops.txt
#chown ansadmin:ansadmin devops.txt

```
[ec2-user@ip-172-31-31-192 test]$ chown ansadmin devops.txt
chown: changing ownership of 'devops.txt': Operation not permitted
[ec2-user@ip-172-31-31-192 test]$ sudo chown ansadmin devops.txt
[ec2-user@ip-172-31-31-192 test]$ ll
total 0
-rw-rw-r--. 1 ansadmin ec2-user 0 Oct 15 09:20 devops.txt
[ec2-user@ip-172-31-31-192 test]$ sudo chown ansadmin:ansadmin devops.txt
[ec2-user@ip-172-31-31-192 test]$ ll
total 0
-rw-rw-r--. 1 ansadmin ansadmin 0 Oct 15 09:20 devops.txt
[ec2-user@ip-172-31-31-192 test]$
```

chgrp : Change file group.

chgrp ec2-user devops.txt

```
[ec2-user@ip-172-31-31-192 test]$ ll
total 0
-rw-rw-r--. 1 ansadmin ansadmin 0 Oct 15 09:20 devops.txt
[ec2-user@ip-172-31-31-192 test]$ sudo chgrp ec2-user devops.txt
[ec2-user@ip-172-31-31-192 test]$ ll
total 0
-rw-rw-r--. 1 ansadmin ec2-user 0 Oct 15 09:20 devops.txt
[ec2-user@ip-172-31-31-192 test]$
```

cp: Copy files and directories.
#cp devops.txtdevops_copy.txt
#cp -r folder1 folder

mv: Rename SOURCE to DEST, or move SOURCE to DIRECTORY/file.
Syntax : mv <<File/ Directory Actual Name>><< File/ Directory New Name>>
#mv devops.txt aws.txt
#mv folder folder4

```
[ec2-user@ip-172-31-31-192 test]$ ll
total 0
-rw-rw-r--. 1 ansadmin ec2-user 0 Oct 15 09:20 devops.txt
[ec2-user@ip-172-31-31-192 test]$ cp devops.txt devops_copy.txt
[ec2-user@ip-172-31-31-192 test]$ ll
total 0
-rw-rw-r--. 1 ec2-user ec2-user 0 Oct 15 09:38 devops_copy.txt
-rw-rw-r--. 1 ansadmin ec2-user 0 Oct 15 09:20 devops.txt
[ec2-user@ip-172-31-31-192 test]$ mv devops.txt aws.txt
[ec2-user@ip-172-31-31-192 test]$ ll
total 0
-rw-rw-r--. 1 ansadmin ec2-user 0 Oct 15 09:20 aws.txt
-rw-rw-r--. 1 ec2-user ec2-user 0 Oct 15 09:38 devops_copy.txt
[ec2-user@ip-172-31-31-192 test]$
```

file: Determine file type.
#file /etc/hosts

```
[ec2-user@ip-172-31-31-192 test]$ file /etc/hosts
/etc/hosts: ASCII text
[ec2-user@ip-172-31-31-192 test]$ file devops_copy.txt
devops_copy.txt: empty
[ec2-user@ip-172-31-31-192 test]$ vi aws.txt
[ec2-user@ip-172-31-31-192 test]$ file aws.txt
aws.txt: ASCII text
[ec2-user@ip-172-31-31-192 test]$
```

```
[ec2-user@ip-172-31-31-192 test]$ cp -r folder1 folder
[ec2-user@ip-172-31-31-192 test]$ ll
total 0
-rw-rw-r--. 1 ansadmin ec2-user 0 Oct 15 09:20 aws.txt
-rw-rw-r--. 1 ec2-user ec2-user 0 Oct 15 09:38 devops_copy.txt
drwxrwxr-x. 2 ec2-user ec2-user 6 Oct 15 09:42 folder
drwxrwxr-x. 2 ec2-user ec2-user 6 Oct 15 09:40 folder1
drwxrwxr-x. 2 ec2-user ec2-user 6 Oct 15 09:40 folder2
drwxrwxr-x. 2 ec2-user ec2-user 6 Oct 15 09:40 folder3
[ec2-user@ip-172-31-31-192 test]$ mv -r folder folder4
mv: invalid option -- 'r'
Try 'mv --help' for more information.
[ec2-user@ip-172-31-31-192 test]$ mv folder folder4
[ec2-user@ip-172-31-31-192 test]$ ll
total 0
-rw-rw-r--. 1 ansadmin ec2-user 0 Oct 15 09:20 aws.txt
-rw-rw-r--. 1 ec2-user ec2-user 0 Oct 15 09:38 devops_copy.txt
drwxrwxr-x. 2 ec2-user ec2-user 6 Oct 15 09:40 folder1
drwxrwxr-x. 2 ec2-user ec2-user 6 Oct 15 09:40 folder2
drwxrwxr-x. 2 ec2-user ec2-user 6 Oct 15 09:40 folder3
drwxrwxr-x. 2 ec2-user ec2-user 6 Oct 15 09:42 folder4
[ec2-user@ip-172-31-31-192 test]$
```

wc: Print the number of newlines,words, and bytes in files Syntax : wc [-c | -m] [-l] [-w] [-L] [File ...]

Options can be

- -c Display number of bytes
- -m Display number of character.
- -l Display number of lines.
- -w Display number of words.
- -L Displays the length of the longest line.

```
[ec2-user@ip-172-31-31-192 test]$ cat aws.txt
Hi this is test file
What users/roles does a Dev EKS/Kubernetes environment cluster generally have, apart from admin & storage admins?
What users/roles does a Dev EKS/Kubernetes environment cluster generally have, apart from admin & storage admins?
What users/roles does a Dev EKS/Kubernetes environment cluster generally have, apart from admin & storage admins?
What users/roles does a Dev EKS/Kubernetes environment cluster generally have, apart from admin & storage admins?

[ec2-user@ip-172-31-31-192 test]$ wc -c aws.txt
479 aws.txt
[ec2-user@ip-172-31-31-192 test]$ wc -l aws.txt
6 aws.txt
[ec2-user@ip-172-31-31-192 test]$ wc -m aws.txt
479 aws.txt
[ec2-user@ip-172-31-31-192 test]$ wc -w aws.txt
69 aws.txt
[ec2-user@ip-172-31-31-192 test]$ wc -L aws.txt
113 aws.txt
[ec2-user@ip-172-31-31-192 test]$
```

ln : Make links between files.

Two types of links:

1. Hard Links
2. Symbolic Links

Hard Links:

In Linux all directories and files have inodes. Creating hard link is nothing but add a new name to the inode. To do this we can use the ln command.

- There is no difference between hard link and the original file.

- Both files have the same inode number.
- Both files have the same size/contents.
- Both files have the same access permissions.
- If you change one file contents it will update another file also

Symbolic links/softlinks:

- The symbolic link and the original file have different inodes.
- The size of the symbolic link is significantly different from the size of the real file.
- The size of the symbolic link is the number of bytes in the name of the file it refers to, because no other information is available in the symbolic link.
- The file type of the symbolic link is set to 1, which indicates that it is a symbolic link.

```
[ec2-user@ip-172-31-31-192 test]$ ln source1.txt hardlink.txt
[ec2-user@ip-172-31-31-192 test]$ ls -li source1.txt hardlink.txt
13246466 -rw-rw-r--. 2 ec2-user ec2-user 0 Oct 15 10:13 hardlink.txt
13246466 -rw-rw-r--. 2 ec2-user ec2-user 0 Oct 15 10:13 source1.txt
[ec2-user@ip-172-31-31-192 test]$ ln -s source2.txt softlink.txt
[ec2-user@ip-172-31-31-192 test]$ ls -li source2.txt softlink.txt
13246468 lrwxrwxrwx. 1 ec2-user ec2-user 11 Oct 15 10:17 softlink.txt -> source2.txt
13246467 -rw-rw-r--. 1 ec2-user ec2-user  0 Oct 15 10:13 source2.txt
[ec2-user@ip-172-31-31-192 test]$
```

What is the difference between Hard Link and Symbolic Link?

A symbolic link is like a shortcut. It points to the original file and helps you find it easily. It breaks if you remove the original file.
A hard ink is like a copy of the original file that is synchronized continuously. There is no difference between the original file and the hard link; they both refer to the same blocks. We can create symbolic link for both files and directories, but we cannot create hard links for directories, can create only for files.

Note: If you create hard links, inode number will be same for both files and if you create soft links, inode number will be different.

```
[ec2-user@ip-172-31-31-192 test]$ mkdir test
[ec2-user@ip-172-31-31-192 test]$ ln test hardtest
ln: test: hard link not allowed for directory
[ec2-user@ip-172-31-31-192 test]$ ln -s test softtest
[ec2-user@ip-172-31-31-192 test]$ ls -li
total 0
13246466 -rw-rw-r--. 2 ec2-user ec2-user  0 Oct 15 10:13 hardlink.txt
13246468 lrwxrwxrwx. 1 ec2-user ec2-user 11 Oct 15 10:17 softlink.txt -> source2.txt
13246469 lrwxrwxrwx. 1 ec2-user ec2-user  4 Oct 15 10:20 softtest -> test
13246466 -rw-rw-r--. 2 ec2-user ec2-user  0 Oct 15 10:13 source1.txt
13246467 -rw-rw-r--. 1 ec2-user ec2-user  0 Oct 15 10:13 source2.txt
 4236678 drwxrwxr-x. 2 ec2-user ec2-user  6 Oct 15 10:19 test
[ec2-user@ip-172-31-31-192 test]$
```

```
[ec2-user@ip-172-31-31-192 test]$ rm source2.txt
[ec2-user@ip-172-31-31-192 test]$ ll
total 8
-rw-rw-r--. 2 ec2-user ec2-user  3 Oct 15 10:21 hardlink.txt
lrwxrwxrwx. 1 ec2-user ec2-user 11 Oct 15 10:17 softlink.txt -> [redacted]
lrwxrwxrwx. 1 ec2-user ec2-user  4 Oct 15 10:20 softtest -> test
-rw-rw-r--. 2 ec2-user ec2-user  3 Oct 15 10:21 source1.txt
drwxrwxr-x. 2 ec2-user ec2-user  6 Oct 15 10:19 test
[ec2-user@ip-172-31-31-192 test]$ rm source1.txt
[ec2-user@ip-172-31-31-192 test]$ ll
total 4
-rw-rw-r--. 1 ec2-user ec2-user  3 Oct 15 10:21 hardlink.txt
lrwxrwxrwx. 1 ec2-user ec2-user 11 Oct 15 10:17 softlink.txt -> [redacted]
lrwxrwxrwx. 1 ec2-user ec2-user  4 Oct 15 10:20 softtest -> test
drwxrwxr-x. 2 ec2-user ec2-user  6 Oct 15 10:19 test
[ec2-user@ip-172-31-31-192 test]$
```

There are two types of Links:-

	Soft Link	Hard link
1	Size of link file is equal to no. of characters in the name of original file	Size of both file is same
2	Can be created across the Partition	Can't be created across the partition
3	Inode no. of source and link file is different	Inode no. of both file is same
4	if original file is deleted, link is broken and data is lost	If original file is deleted then also link will contain data
5	SHORTCUT FILE	BACKUP FILE

vim or vi : Vi improved, a programmers text editor.

#vi psddevops.txt

Then you can enter 'Insert' key or 'i'. Now you are able to type text into that file(Bachi.txt). Once it finishes press 'Esc' button and type ':' and type 'wq'. Option q! is for simply quit without save.

Now check with ls –l command. See here size of the file is 54 bytes.

- Use "^" command to move to beginning of current line.
- Use "$" command to move to end of current line.
- Use "w" command to move forward by word.
- Use "b" command to move back by word.
- Use "e" command to move to end of word.
- Use "Ctrl-F" command to move to next screen.
- Use "Ctrl-D" command to move forward by half a screen.
- Use "Ctrl-B" command to move to previous screen.
- Press "A" command to add data to end of current line.
- Use "yy" command to copy current line.
- Use "<n>yy" command to copy <n> lines from the current line.

- Use “p” to paste the lines cut using the “yy” or the “dd” command. The lines are pasted AFTER the current line.
- Use “P” to paste the lines cut using the “yy” or the “dd” command. The lines are pasted BEFORE the current line.
- Use “.” command to repeat the last add, update, delete or paste command.
- Use “x” command to delete current character.
- Use “<n>x” command to delete <n> characters beginning from the current character.
- Use “r<char>” command to replace current character with <char> character.
- Use “cw” command to change current word with new word(s). Enter new words and press ESC when done.
- Use “<n>cw” command to change <n> words beginning with current word with new word(s). Enter new words and press ESC when done.
- Use “C” command to replace remainder of the line. Enter new text and press ESC when done.
- Use “G” to move to end of file.
- Use “:/string” command to search the “string”.
- Use “n” command to repeat the previous search.
- Use “:s/old_string/new_string” command to substitute old_string with new_string in the current line.
- Use “:<m>,<n>s/old_string/new_string” command to substitute old_string with new_string in line “m” thru “n”.For first line set “m” to “1”. For last line, set “n” to “$”.

echo : Display a line of text.

```
#echo Hi I am Raja, working in wipro Technologies, Electronic city-1,Bangalore.
Hi I am Raja, working in wipro Technologies, Electronic city-1,Bangalore.

#echo "Hi I am Raja, working in wipro Technologies, Electronic city-1,Bangalore."
Hi I am Raja, working in wipro Technologies, Electronic city-1,Bangalore.

#echo Hi          I am Raja, working in wipro Technologies, Electronic city-1,Bangalore.
Hi I am Raja, working in wipro Technologies, Electronic city-1,Bangalore.

#echo "Hi          I am Raja, working in wipro Technologies, Electronic city-1,Bangalore."
Hi          I am Raja, working in wipro Technologies, Electronic city-1,Bangalore.

#echo $PATH
/home/ec2-user/.local/bin:/home/ec2-
user/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin
```

cat: Concatenate files and print on the standard output.

#cat sample.txt
#cat sample.txt |tr a-z A-Z
#cat -n sample.txt

head: Print the first lines of each FILE to standard output.

Syntax: head [OPTION]... [FILE]...
This command is very useful when we want to see some lines in big file.

Options are
1)-n: output the last N lines, instead of the last 10.
#head sample.txt : It will display the first 10 rows for specified file.

Note: By default, it will display 10 rows if don't provide any options for head command.

2) #head -20 sample.txt: It will display the first 20 rows for specified file.
(OR)
#head -n 20 sample.txt

Note: If you specify the more lines than the available in file, it will display the max lines in the file. See in the above example we specified the 20 lines in the command but only 15 lines there in the bhaskar.txt file.

```
[ec2-user@ip-172-31-31-192 ppreddy]$ head sample.txt
What users/roles does a Dev EKS/Kubernetes environment cluster generally
What users/roles does a Dev EKS/Kubernetes environment cluster generally
What users/roles does a Dev EKS/Kubernetes environment cluster generally
What users/roles does a Dev EKS/Kubernetes environment cluster generally
What users/roles does a Dev EKS/Kubernetes environment cluster generally
What users/roles does a Dev EKS/Kubernetes environment cluster generally
What users/roles does a Dev EKS/Kubernetes environment cluster generally
What users/roles does a Dev EKS/Kubernetes environment cluster generally
What users/roles does a Dev EKS/Kubernetes environment cluster generally
What users/roles does a Dev EKS/Kubernetes environment cluster generally
[ec2-user@ip-172-31-31-192 ppreddy]$ head  -3 sample.txt
What users/roles does a Dev EKS/Kubernetes environment cluster generally
What users/roles does a Dev EKS/Kubernetes environment cluster generally
What users/roles does a Dev EKS/Kubernetes environment cluster generally
[ec2-user@ip-172-31-31-192 ppreddy]$ head  -1 sample.txt
What users/roles does a Dev EKS/Kubernetes environment cluster generally
[ec2-user@ip-172-31-31-192 ppreddy]$
```

tail : Print the last lines of each FILE to standard output.

Syntax: tail [OPTION]... [FILE]...

Options are
–f : output appended data as the file grows.
–n : output the last N lines, instead of the last 10.

#tail sample.txt

#tail -3 sample.txt

```
[ec2-user@ip-172-31-31-192 ppreddy]$ tail sample.txt
What users/roles does a Dev EKS/Kubernetes environment cluster generally
What users/roles does a Dev EKS/Kubernetes environment cluster generally
What users/roles does a Dev EKS/Kubernetes environment cluster generally
What users/roles does a Dev EKS/Kubernetes environment cluster generally
What users/roles does a Dev EKS/Kubernetes environment cluster generally
What users/roles does a Dev EKS/Kubernetes environment cluster generally
What users/roles does a Dev EKS/Kubernetes environment cluster generally
What users/roles does a Dev EKS/Kubernetes environment cluster generally
What users/roles does a Dev EKS/Kubernetes environment cluster generally
What users/roles does a Dev EKS/Kubernetes environment cluster generally
[ec2-user@ip-172-31-31-192 ppreddy]$ tail -3 sample.txt
What users/roles does a Dev EKS/Kubernetes environment cluster generally
What users/roles does a Dev EKS/Kubernetes environment cluster generally
What users/roles does a Dev EKS/Kubernetes environment cluster generally
[ec2-user@ip-172-31-31-192 ppreddy]$
```

more: It is a filter for paging through text one screenful at a time (stop the display on each screen)

Syntax: more << file name >>
#morecron

less : less is the same as more except you can scroll back and forward.

Syntax : less << file name >>
#less cron

Following are options with less for navigation.

- CTRL+F – forward one window
- CTRL+B – backward one window
- CTRL+D – forward half window
- CTRL+U – backward half window
- G – go to the end of file
- g – go to the start of file
- q or ZZ – exit the less pager

sort : It is used to sort the output in numeric or alphabetic order

sed : Sed is a Stream editor. A stream editor is used to perform basic text transformations on an input stream (a file or input from a pipeline).
FileName: sedfile.txt
Hi All,
My Name is Raja, working as a DevOps Engineer in Wipro, Bangalore.Currently DevOps is very good in market.
Learning DevOps tools are very easy. DevOps is not a technology.DevOps is a culture.
DevOps is the combination of software development and operations team.DevOps helps an organization deploy software more frequently.
Replacing or substituting string

sed 's/DevOps/Java/' sedfile.txt
Here the ‘s’ represents the substitution operation. ‘/’ is delimeter.
‘DevOps’ is search pattern. ‘Java’ is the substitute string.

Note: By default, the sed command replaces the first occurrence of the pattern in each line and it won't replace the second, third...occurrence in the line.

Replacing the nth occurrence of a pattern in a line.

Use the /1, /2 etc flags to replace the first, second occurrence of a pattern in a line. The below command replaces the second occurrence of the word "unix" with "linux" in a line.

sed 's/DevOps/Java/1' sedfile.txt : It will replace the first occurrence.
sed 's/DevOps/Java/2' sedfile.txt: It will replace the second occurrence.

Replacing all the occurrence of the pattern in a line.
sed 's/DevOps/Java/g' sedfile.txt

‘g’ (global replacement is used to make the changes globally.

awk:
Linux AWK command provides a scripting language for text processing. linux AWK command is simple yet powerful enough to perform text processing based on patterns and rules we specify. It works by scanning the file line by line and performing actions that we have specified when the rules match.

Capabilities of AWK

- Variables
- Arithmatic Operations
- Loops and Control Statements
- Output Formatting
- Pattern matching
- Report Generation

#cat sample.txt
Bill Microsoft
Steve Apple
Elon Tesla
Jeff Amazon

Print the columns with linux AWK command
{print $0} will print all the columns
#awk '{print $0}' sample.txt
{print $1} will print only the first column
#awk '{print $1}' sample.txt
{print $2} will print only the first column

#awk '{print $2}' sample.txt

```
[ec2-user@ip-172-31-31-192 ppreddy]$ awk '{print $0}' sample.txt
Bill Microsoft
Steve Apple
Elon Tesla
Jeff Amazon
[ec2-user@ip-172-31-31-192 ppreddy]$ awk '{print $1}' sample.txt
Bill
Steve
Elon
Jeff
[ec2-user@ip-172-31-31-192 ppreddy]$ awk '{print $2}' sample.txt
Microsoft
Apple
Tesla
Amazon
[ec2-user@ip-172-31-31-192 ppreddy]$
```

Match the Pattern with Linux AWK command

By keeping any pattern in '/' we can match that pattern and it will only print the lines containing the pattern.

#awk '/Bill/ {print $0}' sample.txt

```
[ec2-user@ip-172-31-31-192 ppreddy]$ awk '/Bill/ {print $0}' sample.txt
Bill Microsoft
[ec2-user@ip-172-31-31-192 ppreddy]$
```

Count the rows with linux awk command

NR will count the number of rows in the document.

#awk '{print NR, $0}' sample.txt

```
[ec2-user@ip-172-31-31-192 ppreddy]$ awk '{print NR, $0}' sample.txt
1 Bill Microsoft
2 Steve Apple
3 Elon Tesla
4 Jeff Amazon
[ec2-user@ip-172-31-31-192 ppreddy]$
```

Count the fields with linux awk command

NF will count the number of fields in the record.

#awk '{print NF, $0}' sample.txt

```
[ec2-user@ip-172-31-31-192 ppreddy]$ awk '{print NF, $0}' sample.txt
2 Bill Microsoft
2 Steve Apple
2 Elon Tesla
2 Jeff Amazon
[ec2-user@ip-172-31-31-192 ppreddy]$
```

Seprate Records with linux awk command

We can use ORS (Output Record Seprator) to seprate records using desired character with awk command.

#awk 'BEGIN {ORS=":"} {print $0}' sample.txt

```
[ec2-user@ip-172-31-31-192 ppreddy]$ awk 'BEGIN {ORS=":"} {print $0}' sample.txt
Bill Microsoft:Steve Apple:Elon Tesla:Jeff Amazon:[ec2-user@ip-172-31-31-192 ppreddy]$
```

We can use OFS (Output Field Seprator) to seprate fields using the desired character.
#awk 'BEGIN {OFS="="} {print $1, $2}' sample.txt

```
[ec2-user@ip-172-31-31-192 ppreddy]$ awk 'BEGIN {OFS="="} {print $1, $2}' sample.txt
Bill=Microsoft
Steve=Apple
Elon=Tesla
Jeff=Amazon
[ec2-user@ip-172-31-31-192 ppreddy]$
```

Count The Records with linux awk command

We can count the number of records using NR in awk command
#awk 'END {print NR}' sample.txt

```
[ec2-user@ip-172-31-31-192 ppreddy]$ awk 'END {print NR}' sample.txt
4
[ec2-user@ip-172-31-31-192 ppreddy]$
```

Use Loops in Linux AWK command

We can use for loop in linux awk command.
#awk 'BEGIN {for(i=1;i<=5;i++) print "Hello ",i }'

```
[ec2-user@ip-172-31-31-192 ppreddy]$ awk 'BEGIN {for(i=1;i<=5;i++) print "Hello ",i }'
Hello  1
Hello  2
Hello  3
Hello  4
Hello  5
[ec2-user@ip-172-31-31-192 ppreddy]$
```

Use If statement in linux awk command

We can use control statements like ‘if statement’ in awk command as below.
#awk '{if($2=="Apple") print $0}' sample.txt

```
[ec2-user@ip-172-31-31-192 ppreddy]$ awk '{if($2=="Apple") print $0}' sample.txt
Steve Apple
[ec2-user@ip-172-31-31-192 ppreddy]$
```

grep: This command is used to search specific string in specified file. By default, grep prints the matching lines.

#grep DevOps *.*
#grep -i DevOps *.*
#grep DevOps test.txt
#grep -i DevOps test.txt: Ignore case distinctions in both the PATTERN.
#grep -c -i DevOps test.txt: With –c option it will display how many lines matches the given pattern/string.

```
[ec2-user@ip-172-31-31-192 ppreddy]$ grep -i DevOps *.*
test.txt:Hi All,My Name is Raja, working as a DevOps Engineer in Wipro, Bangalore.Currently DevOps is very good in market.
test.txt:Learning DevOps tools are very easy. DevOps is not a technology.DevOps is a culture.
test.txt:DevOps is the combination of software development and operations team. DevOps helps an organization deploy softwa
re
[ec2-user@ip-172-31-31-192 ppreddy]$ grep -i DevOps test.txt
Hi All,My Name is Raja, working as a DevOps Engineer in Wipro, Bangalore.Currently DevOps is very good in market.
Learning DevOps tools are very easy. DevOps is not a technology.DevOps is a culture.
DevOps is the combination of software development and operations team. DevOps helps an organization deploy software
[ec2-user@ip-172-31-31-192 ppreddy]$ grep -c -i DevOps test.txt
3
[ec2-user@ip-172-31-31-192 ppreddy]$ grep -i "DevOps Engineer" test.txt
Hi All,My Name is Raja, working as a DevOps Engineer in Wipro, Bangalore.Currently DevOps is very good in market.
[ec2-user@ip-172-31-31-192 ppreddy]$
```

#ps -ef |grep ec2-user: It will display all the ec2-user process.

```
[ec2-user@ip-172-31-31-192 ppreddy]$ ps -ef |grep ec2-user
root        1285     961  0 01:36 ?        00:00:00 sshd: ec2-user [priv]
root        1287     961  0 01:36 ?        00:00:00 sshd: ec2-user [priv]
ec2-user    1290       1  0 01:36 ?        00:00:00 /usr/lib/systemd/systemd --user
ec2-user    1294    1290  0 01:36 ?        00:00:00 (sd-pam)
ec2-user    1300    1285  0 01:36 ?        00:00:00 sshd: ec2-user@pts/0
ec2-user    1302    1300  0 01:37 pts/0    00:00:00 -bash
ec2-user    1325    1287  0 01:37 ?        00:00:00 sshd: ec2-user@notty
ec2-user    1327    1325  0 01:37 ?        00:00:00 /usr/libexec/openssh/sftp-server
ec2-user    1370    1302  0 01:47 pts/0    00:00:00 ps -ef
ec2-user    1371    1302  0 01:47 pts/0    00:00:00 grep --color=auto ec2-user
[ec2-user@ip-172-31-31-192 ppreddy]$
```

who: It shows who is logged in the same machine.

#who -T :
#who -H :
#who am i :
The use of the arguments with who am i causes the command to only lists the one user who typed the command.The command is who and the arguments are am i.

w : Show who is logged on and what they are doing.

The header shows, in this order, the current time, how long the system has been running, how many users are currently logged on, and the system load averages for the past 1, 5, and 15 minutes.
The following entries are displayed for each user: login name, the tty name, the remote host, login time, idle time, JCPU, PCPU, and the command line of their current process.
The JCPU time is the time used by all processes attached to the tty. It does not include past background jobs, but does include currently running back- ground jobs.
The PCPU time is the time used by the current process, named in the "what" field.

e.g. #w

#whoami : Print the user name associated with the current effective user ID. Same as id -un.

#id -un :

```
[ec2-user@ip-172-31-31-192 ppreddy]$ who
ec2-user pts/0        2020-10-16 01:37 (106.206.53.170)
[ec2-user@ip-172-31-31-192 ppreddy]$ who -T
ec2-user + pts/0        2020-10-16 01:37 (106.206.53.170)
[ec2-user@ip-172-31-31-192 ppreddy]$ who -H
NAME     LINE         TIME             COMMENT
ec2-user pts/0        2020-10-16 01:37 (106.206.53.170)
[ec2-user@ip-172-31-31-192 ppreddy]$ whoami
ec2-user
[ec2-user@ip-172-31-31-192 ppreddy]$ w
 01:49:51 up 15 min,  1 user,  load average: 0.00, 0.01, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
ec2-user pts/0    106.206.53.170   01:37    0.00s  0.02s  0.00s w
[ec2-user@ip-172-31-31-192 ppreddy]$ who am i
ec2-user pts/0        2020-10-16 01:37 (106.206.53.170)
[ec2-user@ip-172-31-31-192 ppreddy]$ id -un
ec2-user
[ec2-user@ip-172-31-31-192 ppreddy]$
```

whereis: Path/locate the binary, source, and manual page files for a command.

Syntax: whereis << options >><< command >> Options are:

- -bSearch only for binaries.
- -m Search only for manual sections.
- -s Search only for sources.

Examples:

whereis chmod
whereis -b chmod ---> Gives the path for binaries.
whereis -m chmod ---> Gives the path for manual sections.
whereis -s chmod ---> Gives the path for sources.

```
[ec2-user@ip-172-31-31-192 ~]$ whereis chmod
chmod: /usr/bin/chmod /usr/share/man/man1/chmod.1.gz
[ec2-user@ip-172-31-31-192 ~]$ whereis -b chmod
chmod: /usr/bin/chmod
[ec2-user@ip-172-31-31-192 ~]$ whereis -m chmod
chmod: /usr/share/man/man1/chmod.1.gz
[ec2-user@ip-172-31-31-192 ~]$ whereis -s chmod
chmod:
[ec2-user@ip-172-31-31-192 ~]$
```

date: Displays or sets the date or time.

Syntax : date +[option]

The options can be

- %D-displays date as mm/dd/yy
- %a-displays abbreviated weekdays
- %h-displays abbreviated month
- %m-displays month of year
- %d-displays day of the month
- %y-displays last 2 digits of the month
- %T-displays time as HH:MM:SS

- %H-displays hour as between 00 to 23
- %M-displays the minute between 00 to 59
- %S-displays second as between 00 to 59

Ex:

date +%D
date +%a
date +%h
date +%m
date +%d
date +%y
date +%T
date +%H
date +%M
date +%S

```
[ec2-user@ip-172-31-31-192 ~]$ date +%D
10/16/20
[ec2-user@ip-172-31-31-192 ~]$ date +%a
Fri
[ec2-user@ip-172-31-31-192 ~]$ date +%h
Oct
[ec2-user@ip-172-31-31-192 ~]$ date +%m
10
[ec2-user@ip-172-31-31-192 ~]$ date +%d
16
[ec2-user@ip-172-31-31-192 ~]$ date +%y
20
[ec2-user@ip-172-31-31-192 ~]$ date +%T
02:05:32
[ec2-user@ip-172-31-31-192 ~]$ date +%H
02
[ec2-user@ip-172-31-31-192 ~]$ date +%M
05
[ec2-user@ip-172-31-31-192 ~]$ date +%S
32
```

#date +%Y%m%d -s "19840504"
#date -s "05 April 1984 04:00:00"
#date +%T -s "10:12:12"
#date +%T%p -s "12:54:24 AM"
%p locale's equivalent of either AM or PM

```
[ec2-user@ip-172-31-31-192 ~]$ sudo date +%Y%m%d -s "19840504"
19840504
[ec2-user@ip-172-31-31-192 ~]$ date
Fri May  4 00:00:11 UTC 1984
[ec2-user@ip-172-31-31-192 ~]$ sudo date -s "05 April 1984 04:00:00"
Thu Apr  5 04:00:00 UTC 1984
[ec2-user@ip-172-31-31-192 ~]$ sudo date +%T -s "10:12:12"
10:12:12
[ec2-user@ip-172-31-31-192 ~]$ date
Thu Apr  5 10:12:16 UTC 1984
[ec2-user@ip-172-31-31-192 ~]$ date +%T%p -s "12:54:24 AM"
date: cannot set date: Operation not permitted
00:54:24AM
[ec2-user@ip-172-31-31-192 ~]$ sudo date +%T%p -s "12:54:24 AM"
00:54:24AM
[ec2-user@ip-172-31-31-192 ~]$
```

df command

The df command (short for disk free), is used to display information related to file systems about total space and available space.

Syntax:

df [OPTION]... [FILE]... OPTIONS

Show information about the file system on which each FILE resides, or all file systems by default.

-a, --all include dummy file systems

-B, --block-size=SIZE use SIZE-byte blocks

-h, --human-readable print sizes in human readable format (e.g., 1K 234M 2G)

-H, --si likewise, but use powers of 1000 not 1024

-i, --inodes list inode information instead of block usage

-k, --like --block-size=1K

-l, --local limit listing to local file systems

--no-sync do not invoke sync before getting usage info (default)

-P, --portability use the POSIX output format

--sync invoke sync before getting usage info

-t, --type=TYPE limit listing to file systems of type TYPE

-T, --print-type print file system type

-x, --exclude-type=TYPE limit listing to file systems not of type TYPE

--help display this help and exit

--version
output version information and exit

SIZE may be (or may be an integer optionally followed by) one of following: KB 1000, K 1024, MB 1000*1000, M 1024*1024, and so on for G, T, P, E, Z, Y.

If you want to display all the file system, use -a option.

#df -a

```
[ec2-user@ip-172-31-31-192 ~]$ df -a
Filesystem     1K-blocks    Used Available Use% Mounted on
sysfs                  0       0         0    - /sys
proc                   0       0         0    - /proc
devtmpfs          394996       0    394996   0% /dev
securityfs             0       0         0    - /sys/kernel/security
tmpfs             417784       0    417784   0% /dev/shm
devpts                 0       0         0    - /dev/pts
tmpfs             417784   31796    385988   8% /run
tmpfs             417784       0    417784   0% /sys/fs/cgroup
cgroup                 0       0         0    - /sys/fs/cgroup/systemd
pstore                 0       0         0    - /sys/fs/pstore
bpf                    0       0         0    - /sys/fs/bpf
cgroup                 0       0         0    - /sys/fs/cgroup/rdma
cgroup                 0       0         0    - /sys/fs/cgroup/memory
cgroup                 0       0         0    - /sys/fs/cgroup/net_cls,net_prio
cgroup                 0       0         0    - /sys/fs/cgroup/blkio
cgroup                 0       0         0    - /sys/fs/cgroup/pids
cgroup                 0       0         0    - /sys/fs/cgroup/freezer
cgroup                 0       0         0    - /sys/fs/cgroup/cpuset
cgroup                 0       0         0    - /sys/fs/cgroup/cpu,cpuacct
cgroup                 0       0         0    - /sys/fs/cgroup/devices
cgroup                 0       0         0    - /sys/fs/cgroup/perf_event
cgroup                 0       0         0    - /sys/fs/cgroup/hugetlb
configfs               0       0         0    - /sys/kernel/config
/dev/xvda2      10473452 1205500   9267952  12% /
selinuxfs              0       0         0    - /sys/fs/selinux
systemd-1              -       -         -    - /proc/sys/fs/binfmt_misc
hugetlbfs              0       0         0    - /dev/hugepages
debugfs                0       0         0    - /sys/kernel/debug
mqueue                 0       0         0    - /dev/mqueue
tmpfs              83556       0     83556   0% /run/user/1000
binfmt_misc            0       0         0    - /proc/sys/fs/binfmt_misc
[ec2-user@ip-172-31-31-192 ~]$
```

Use -h option to display size in power of 1024

#df -h

```
[ec2-user@ip-172-31-31-192 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        386M     0  386M   0% /dev
tmpfs           408M     0  408M   0% /dev/shm
tmpfs           408M   32M  377M   8% /run
tmpfs           408M     0  408M   0% /sys/fs/cgroup
/dev/xvda2       10G  1.2G  8.9G  12% /
tmpfs            82M     0   82M   0% /run/user/1000
[ec2-user@ip-172-31-31-192 ~]$
```

Use -H option to display sizes in power of 1000

#df –H

```
[ec2-user@ip-172-31-31-192 ~]$ df -H
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        405M     0  405M   0% /dev
tmpfs           428M     0  428M   0% /dev/shm
tmpfs           428M   33M  396M   8% /run
tmpfs           428M     0  428M   0% /sys/fs/cgroup
/dev/xvda2       11G  1.3G  9.5G  12% /
tmpfs            86M     0   86M   0% /run/user/1000
[ec2-user@ip-172-31-31-192 ~]$ df -H /home/ec2-user/
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda2       11G  1.3G  9.5G  12% /
[ec2-user@ip-172-31-31-192 ~]$ df -h /home/ec2-user/
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda2       10G  1.2G  8.9G  12% /
[ec2-user@ip-172-31-31-192 ~]$
```

To get complete grand total, use –total option

#df –total

```
[ec2-user@ip-172-31-31-192 ~]$ df --total
Filesystem     1K-blocks    Used Available Use% Mounted on
devtmpfs          394996       0    394996   0% /dev
tmpfs             417784       0    417784   0% /dev/shm
tmpfs             417784   31796    385988   8% /run
tmpfs             417784       0    417784   0% /sys/fs/cgroup
/dev/xvda2      10473452 1206016   9267436  12% /
tmpfs              83556       0     83556   0% /run/user/1000
total           12205356 1237812  10967544  11% -
```

Use -T option to display file type

df -T /home/ec2-user/

```
[ec2-user@ip-172-31-31-192 ~]$ df -T /home/ec2-user/
Filesystem     Type 1K-blocks    Used Available Use% Mounted on
/dev/xvda2     xfs   10473452 1205084   9268368  12% /
[ec2-user@ip-172-31-31-192 ~]$
```

du command

du command, short for disk usage, is used to estimate file space usage.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]... [FILE]...
du [OPTION]... --files0-from=F

Options

- -0, –null : end each output line with NULL
- -a, –all : write count of all files, not just directories
- –apparent-size : print apparent sizes, rather than disk usage.
- -B, –block-size=SIZE : scale sizes to SIZE before printing on console
- -c, –total : produce grand total
- -d, –max-depth=N : print total for directory only if it is N or fewer levels below command line argument
- -h, –human-readable : print sizes in human readable format
- -S, -separate-dirs : for directories, don't include size of subdirectories
- -s, –summarize : display only total for each directory
- –time : show time of last modification of any file or directory.
- –exclude=PATTERN : exclude files that match PATTERN

If we want to print sizes in human readable format(K, M, G), use -h option

#du -h /home/ec2-user/ppreddy/
#du -h /home/ec2-user

```
[ec2-user@ip-172-31-31-192 ~]$ du -h /home/ec2-user/ppreddy/
4.0K    /home/ec2-user/ppreddy/sample
16K     /home/ec2-user/ppreddy/
[ec2-user@ip-172-31-31-192 ~]$ du -h /home/ec2-user
4.0K    /home/ec2-user/.ssh
4.0K    /home/ec2-user/ppreddy/sample
16K     /home/ec2-user/ppreddy
0       /home/ec2-user/folder1/test
0       /home/ec2-user/folder1/rootdir
0       /home/ec2-user/folder1
0       /home/ec2-user/folder2
0       /home/ec2-user/devops_batch
0       /home/ec2-user/700
0       /home/ec2-user/perm.txt
0       /home/ec2-user/sample_folder
0       /home/ec2-user/dir/dir1/dir2/dir3
0       /home/ec2-user/dir/dir1/dir2
0       /home/ec2-user/dir/dir1
0       /home/ec2-user/dir
0       /home/ec2-user/display
0       /home/ec2-user/test/test
4.0K    /home/ec2-user/test
56K     /home/ec2-user
[ec2-user@ip-172-31-31-192 ~]$
```

Use -a option for printing all files including directories.

#du -a /home/ec2-user/ppreddy/

```
[ec2-user@ip-172-31-31-192 ~]$ du -a /home/ec2-user/ppreddy/
4       /home/ec2-user/ppreddy/.mytext.csv
4       /home/ec2-user/ppreddy/sample/hardlink.txt
0       /home/ec2-user/ppreddy/sample/softlink.txt
4       /home/ec2-user/ppreddy/sample
4       /home/ec2-user/ppreddy/sample.txt
4       /home/ec2-user/ppreddy/test.txt
16      /home/ec2-user/ppreddy/
[ec2-user@ip-172-31-31-192 ~]$
```

Use -c option to print total size

#du -c /home/ec2-user/ppreddy/

```
[ec2-user@ip-172-31-31-192 ~]$ du -c /home/ec2-user/ppreddy/
4       /home/ec2-user/ppreddy/sample
16      /home/ec2-user/ppreddy/
16      total
[ec2-user@ip-172-31-31-192 ~]$
```

To print sizes till particular level, use -d option with level no.

#du -d 1 /home/ec2-user/

#du -d 2 /home/ec2-user/

```
[ec2-user@ip-172-31-31-192 ~]$ du -d 1 /home/ec2-user/
4       /home/ec2-user/.ssh
16      /home/ec2-user/ppreddy
0       /home/ec2-user/folder1
0       /home/ec2-user/folder2
0       /home/ec2-user/devops_batch
0       /home/ec2-user/700
0       /home/ec2-user/perm.txt
0       /home/ec2-user/sample_folder
0       /home/ec2-user/dir
0       /home/ec2-user/display
4       /home/ec2-user/test
56      /home/ec2-user/
[ec2-user@ip-172-31-31-192 ~]$ du -d 2 /home/ec2-user/
4       /home/ec2-user/.ssh
4       /home/ec2-user/ppreddy/sample
16      /home/ec2-user/ppreddy
0       /home/ec2-user/folder1/test
0       /home/ec2-user/folder1/rootdir
0       /home/ec2-user/folder1
0       /home/ec2-user/folder2
0       /home/ec2-user/devops_batch
0       /home/ec2-user/700
0       /home/ec2-user/perm.txt
0       /home/ec2-user/sample_folder
0       /home/ec2-user/dir/dir1
0       /home/ec2-user/dir
0       /home/ec2-user/display
0       /home/ec2-user/test/test
4       /home/ec2-user/test
56      /home/ec2-user/
[ec2-user@ip-172-31-31-192 ~]$
```

Get summary of file system using -s option

#du -s /home/ec2-user/ppreddy/

```
[ec2-user@ip-172-31-31-192 ~]$ du -s /home/ec2-user/ppreddy/
16      /home/ec2-user/ppreddy/
[ec2-user@ip-172-31-31-192 ~]$
```

Get the timestamp of last modified using --time option

#du --time -h /home/ec2-user/ppreddy/

```
[ec2-user@ip-172-31-31-192 ~]$ du --time -h /home/ec2-user/ppreddy/
4.0K    2020-10-15 13:52        /home/ec2-user/ppreddy/sample
16K     2020-10-16 01:40        /home/ec2-user/ppreddy/
[ec2-user@ip-172-31-31-192 ~]$
```

hostname: Show or set the systems host name.

#hostname
ip-172-31-31-192.us-east-2.compute.internal
#hostname ppreddy.com
#hostname
ppreddy.com
Once the hostname is changed, verify that it has changed the hostname successfully. As you see below, it has changed the hostname to ppreddy.com

```
[ec2-user@ip-172-31-31-192 ~]$ hostname
ip-172-31-31-192.us-east-2.compute.internal
[ec2-user@ip-172-31-31-192 ~]$ hostname ppreddy.com
hostname: you must be root to change the host name
[ec2-user@ip-172-31-31-192 ~]$ sudo hostname ppreddy.com
[ec2-user@ip-172-31-31-192 ~]$ hostname
ppreddy.com
[ec2-user@ip-172-31-31-192 ~]$
```

If the hostname is not changed in /etc/hosts file, need to modify as follows.
#vi /etc/hostname
#vi /etc/hosts
#vi /etc/sysconfig/network

Restart the server as follows. #service network restart

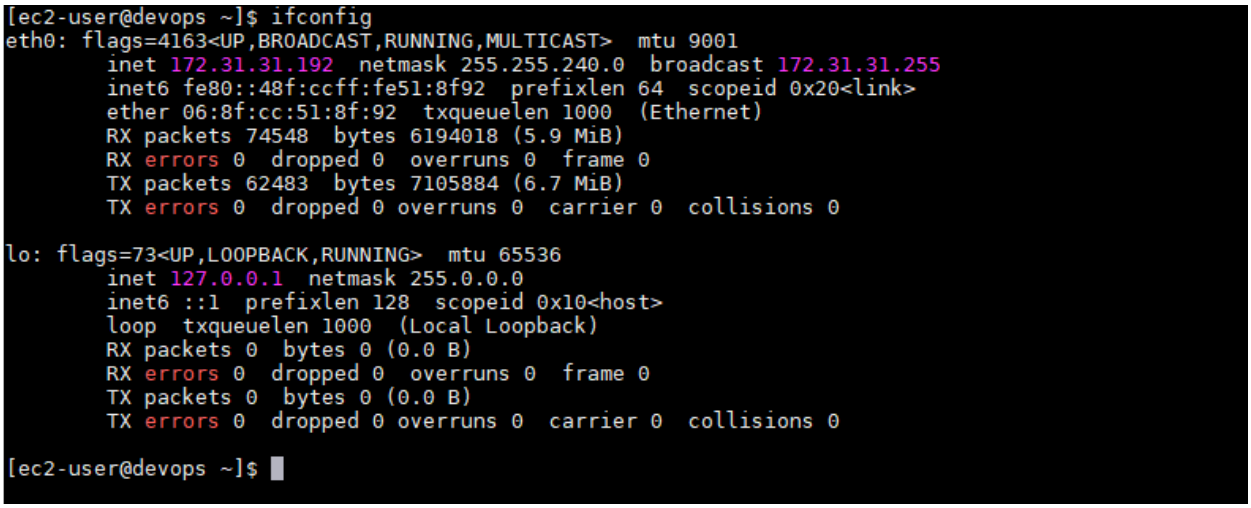
How to find the IP address?

We can find the IP address as follows.

ifconfig: The "ifconfig" (interface configurator) command is used to setup network interfaces and allow the user to view information about the configured network interfaces.

To check the ip address assign to all the interfaces

#ifconfig



To chech the ip of a particular interface

#ifconfig < adapter name >
#ifconfig eth0

```
[ec2-user@devops ~]$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
        inet 172.31.31.192  netmask 255.255.240.0  broadcast 172.31.31.255
        inet6 fe80::48f:ccff:fe51:8f92  prefixlen 64  scopeid 0x20<link>
        ether 06:8f:cc:51:8f:92  txqueuelen 1000  (Ethernet)
        RX packets 74624  bytes 6198626 (5.9 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 62546  bytes 7111268 (6.7 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[ec2-user@devops ~]$ ifconfig lo
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[ec2-user@devops ~]$
```

To check ip of the host

hostname –i

```
[ec2-user@devops ~]$ hostname -i
35.185.75.107
[ec2-user@devops ~]$
```

To check the public ip of the ec2 instance

#curl -s ifconfig.co

#curl -s v4.ident.me

```
[ec2-user@devops ~]$ curl -s v4.ident.me
3.135.222.197[ec2-user@devops ~]$
[ec2-user@devops ~]$
[ec2-user@devops ~]$ curl -s ifconfig.co
3.135.222.197
[ec2-user@devops ~]$
```

To check whether DNS is resolving or not

#host < ip address >

#host 3.135.222.197

#host <hostname>

#host devops.com

#nslookup < ip address >

#nslookup 3.135.222.197

#nslookup < hostname >

#nslookup devops.com

```
[ec2-user@devops ~]$ host 3.135.222.197
197.222.135.3.in-addr.arpa domain name pointer ec2-3-135-222-197.us-east-2.compute.amazonaws.com.
[ec2-user@devops ~]$ host devops.com
devops.com has address 35.185.75.107
devops.com mail is handled by 20 ALT1.ASPMX.L.GOOGLE.com.
devops.com mail is handled by 30 ALT2.ASPMX.L.GOOGLE.com.
devops.com mail is handled by 40 ASPMX2.GOOGLEMAIL.com.
devops.com mail is handled by 50 ASPMX3.GOOGLEMAIL.com.
devops.com mail is handled by 10 ASPMX.L.GOOGLE.com.
[ec2-user@devops ~]$ nslookup 3.135.222.197
197.222.135.3.in-addr.arpa      name = ec2-3-135-222-197.us-east-2.compute.amazonaws.com.

Authoritative answers can be found from:

[ec2-user@devops ~]$ nslookup devops.com
Server:         172.31.0.2
Address:        172.31.0.2#53

Non-authoritative answer:
Name:   devops.com
Address: 35.185.75.107

[ec2-user@devops ~]$
```

man : Displays manual entries online.(To get help of the commands.)

Syntax : man <<Type any command>>

e.g. man man
man who

To search any word in “man” page use “/” and string name. “n” for checking next search and “N” for previous search.
To quit press “:q”.

```
[ec2-user@devops ~]$ man man
[ec2-user@devops ~]$ man who
[ec2-user@devops ~]$ man chmod
[ec2-user@devops ~]$
```

What is the differance between Ctrl +c and Ctrl + Z?
Ctrl + c : It will kill the process with a signal SIGINT.
Ctrl + z : It will suspend the process with a signal SIGSTOP.

However when a process is suspended , we can resume it again by fg (resume in forground) and bg (resume in background) ,
but i cant resume a killed process , that is a difference between using Ctrl+C & Ctrl+Z. By Using 'jobs' command we can see the suspended jobs.
By using fg command we can run the suspended job in foreground and using bg command we can run the suspended job in backroundas follows.
fg --> If only one job is suspended. (OR)
fg %n --> If more than one job is suspended, we will use this command, here 'n' is the job number.
bg
(OR)
bg %n

By using kill command kill the suspended process as follows.

kill %n --> where n will be numbers displayed in jobs command , so if you want to kill 1st process use below command.
kill %1 --> It will kill the first job

info: Gives the information about any command.
Syntax: info <<Type any command>>
Ex:
#info man
#info who

```
[ec2-user@devops ~]$ info man
[ec2-user@devops ~]$ info who
[ec2-user@devops ~]$
```

help:
It gives a short explanation about how to use the command and a list of available options.
Syntax : <<Type any command>> --help
Ex:
#ls --help
#passwd –help

MANAGING INSTALLED SERVICES

- Services are programs (called daemons) that once started run continuously in the background and are ready for input or monitor changes in your computer and respond to them. For example the Apache server has a daemon called httpd (the d is for daemon) that listens on port 80 on your computer and when it receives a request for a page it sends the appropriate data back to the client machine.
- Many services are required to run all the time however many can be safely turned of for both security reasons as running unnecessary services opens more doors into your computer, but also for performance reasons. It may not make much difference but your computer should boot slightly faster with less services it has to start on boot.
- One of the techniques in every Linux administrator's toolbox to improve security of a box is to turn off unneeded services.

chkconfig and service commands

There are 2 commands used to control services.

Service
This controls the starting and stopping of services during a session, these setting are not saved. If you start Apache this way but it is not set to start on boot using the above method then it will continue to run but on next boot will not start automatically.

Chkconfig

This controls which services are set to start on boot, by their nature these setting are saved and are applied at next boot. Changing these settings will not start the service immediately; it will just flag them to be started from the next boot.

The command use for maintaining a service is

#service <name of the service> status --- To check the status of the service
#service <name of the service> start --- To start the service
#service <name of the service > stop --- To stop a service
#service <name of the service> reload --- To reload the service
#service <name of the service> restart --- To restart the service

The command use for service availability is

#chkconfig - -list --- To check the availability of service
#chkconfig <service> on --- To make the service available after restart
#chkconfig <service> off --- To make the service unavailable after restart

#service crond status
#service crond stop
#service crond start
#service crond restart

```
[root@devops systemd]# service crond status
Redirecting to /bin/systemctl status crond.service
● crond.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2020-10-16 01:34:47 UTC; 1 day 15h ago
 Main PID: 969 (crond)
    Tasks: 1 (limit: 4936)
   Memory: 2.7M
   CGroup: /system.slice/crond.service
           └─969 /usr/sbin/crond -n

Oct 17 08:01:01 devops.com CROND[6336]: (root) CMD (run-parts /etc/cron.hourly)
Oct 17 09:01:01 devops.com CROND[6411]: (root) CMD (run-parts /etc/cron.hourly)
Oct 17 10:01:01 devops.com CROND[6480]: (root) CMD (run-parts /etc/cron.hourly)
Oct 17 11:01:01 devops.com CROND[6565]: (root) CMD (run-parts /etc/cron.hourly)
Oct 17 12:01:01 devops.com CROND[6745]: (root) CMD (run-parts /etc/cron.hourly)
Oct 17 13:01:01 devops.com CROND[6809]: (root) CMD (run-parts /etc/cron.hourly)
Oct 17 14:01:01 devops.com CROND[6874]: (root) CMD (run-parts /etc/cron.hourly)
Oct 17 15:01:01 devops.com CROND[6935]: (root) CMD (run-parts /etc/cron.hourly)
Oct 17 16:01:01 devops.com CROND[7088]: (root) CMD (run-parts /etc/cron.hourly)
Oct 17 17:01:01 devops.com CROND[7720]: (root) CMD (run-parts /etc/cron.hourly)
[root@devops systemd]#
```

```
[root@devops systemd]# service crond stop
Redirecting to /bin/systemctl stop crond.service
[root@devops systemd]# service crond status
Redirecting to /bin/systemctl status crond.service
● crond.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Sat 2020-10-17 17:21:40 UTC; 6s ago
  Process: 969 ExecStart=/usr/sbin/crond -n $CRONDARGS (code=exited, status=0/SUCCESS)
 Main PID: 969 (code=exited, status=0/SUCCESS)

Oct 17 10:01:01 devops.com CROND[6480]: (root) CMD (run-parts /etc/cron.hourly)
Oct 17 11:01:01 devops.com CROND[6565]: (root) CMD (run-parts /etc/cron.hourly)
Oct 17 12:01:01 devops.com CROND[6745]: (root) CMD (run-parts /etc/cron.hourly)
Oct 17 13:01:01 devops.com CROND[6809]: (root) CMD (run-parts /etc/cron.hourly)
Oct 17 14:01:01 devops.com CROND[6874]: (root) CMD (run-parts /etc/cron.hourly)
Oct 17 15:01:01 devops.com CROND[6935]: (root) CMD (run-parts /etc/cron.hourly)
Oct 17 16:01:01 devops.com CROND[7088]: (root) CMD (run-parts /etc/cron.hourly)
Oct 17 17:01:01 devops.com CROND[7720]: (root) CMD (run-parts /etc/cron.hourly)
Oct 17 17:21:40 devops.com systemd[1]: Stopping Command Scheduler...
Oct 17 17:21:40 devops.com systemd[1]: Stopped Command Scheduler.
[root@devops systemd]#
```

```
[root@devops systemd]# service crond start
Redirecting to /bin/systemctl start crond.service
[root@devops systemd]# service crond status
Redirecting to /bin/systemctl status crond.service
● crond.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2020-10-17 17:23:30 UTC; 8s ago
 Main PID: 8495 (crond)
    Tasks: 1 (limit: 4936)
   Memory: 1.0M
   CGroup: /system.slice/crond.service
           └─8495 /usr/sbin/crond -n

Oct 17 17:23:30 devops.com systemd[1]: Started Command Scheduler.
Oct 17 17:23:30 devops.com crond[8495]: (CRON) STARTUP (1.5.2)
Oct 17 17:23:30 devops.com crond[8495]: (CRON) INFO (Syslog will be used instead of sendmail.)
Oct 17 17:23:30 devops.com crond[8495]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 67% if used.)
Oct 17 17:23:30 devops.com crond[8495]: (CRON) INFO (running with inotify support)
Oct 17 17:23:30 devops.com crond[8495]: (CRON) INFO (@reboot jobs will be run at computer's startup.)
[root@devops systemd]#
```

```
[root@devops systemd]# service crond restart
Redirecting to /bin/systemctl restart crond.service
[root@devops systemd]# service crond status
Redirecting to /bin/systemctl status crond.service
● crond.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2020-10-17 17:25:19 UTC; 4s ago
 Main PID: 8543 (crond)
    Tasks: 1 (limit: 4936)
   Memory: 1.0M
   CGroup: /system.slice/crond.service
           └─8543 /usr/sbin/crond -n

Oct 17 17:25:19 devops.com systemd[1]: Started Command Scheduler.
Oct 17 17:25:19 devops.com crond[8543]: (CRON) STARTUP (1.5.2)
Oct 17 17:25:19 devops.com crond[8543]: (CRON) INFO (Syslog will be used instead of sendmail.)
Oct 17 17:25:19 devops.com crond[8543]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 8% if used.)
Oct 17 17:25:19 devops.com crond[8543]: (CRON) INFO (running with inotify support)
Oct 17 17:25:19 devops.com crond[8543]: (CRON) INFO (@reboot jobs will be run at computer's startup.)
[root@devops systemd]#
```

ps: This command can used to display the currently running processes on Linux systems.
ps is the shortage for Process Status.

ps:
It will display the user's currently running processes
#ps -f:
Full listning of the user's currently running processes

Prepared by PPNREDDY .....

```
[ec2-user@samplegoogle ~]$ ps
    PID TTY          TIME CMD
   1337 pts/1    00:00:00 bash
   1374 pts/1    00:00:00 ps
[ec2-user@samplegoogle ~]$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
ec2-user    1337    1334  0 04:18 pts/1    00:00:00 -bash
ec2-user    1375    1337  0 04:20 pts/1    00:00:00 ps -f
```

#ps -ef:

Full listing of all processes, except kernel processes.

```
[ec2-user@samplegoogle ~]$ ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 04:11 ?        00:00:02 /usr/lib/systemd/systemd --switched-root --system --deserialize 16
root           2       0  0 04:11 ?        00:00:00 [kthreadd]
root           3       2  0 04:11 ?        00:00:00 [rcu_gp]
root           4       2  0 04:11 ?        00:00:00 [rcu_par_gp]
root           6       2  0 04:11 ?        00:00:00 [kworker/0:0H-kblockd]
root           7       2  0 04:11 ?        00:00:00 [kworker/u30:0-events_unbound]
root           8       2  0 04:11 ?        00:00:00 [mm_percpu_wq]
root           9       2  0 04:11 ?        00:00:00 [ksoftirqd/0]
root          10       2  0 04:11 ?        00:00:00 [rcu_sched]
root          11       2  0 04:11 ?        00:00:00 [migration/0]
root          12       2  0 04:11 ?        00:00:00 [watchdog/0]
root          13       2  0 04:11 ?        00:00:00 [cpuhp/0]
root          15       2  0 04:11 ?        00:00:00 [kdevtmpfs]
root          16       2  0 04:11 ?        00:00:00 [netns]
root          17       2  0 04:11 ?        00:00:00 [kauditd]
root          18       2  0 04:11 ?        00:00:00 [xenbus]
root          19       2  0 04:11 ?        00:00:00 [xenwatch]
root          21       2  0 04:11 ?        00:00:00 [khungtaskd]
root          22       2  0 04:11 ?        00:00:00 [oom_reaper]
root          23       2  0 04:11 ?        00:00:00 [writeback]
```

#ps -A (OR) ps -Af: All processes, including kernel processes.

Here f means full listing. You can observer output without f option and with f option.

```
[ec2-user@samplegoogle ~]$ ps -A
    PID TTY          TIME CMD
      1 ?        00:00:02 systemd
      2 ?        00:00:00 kthreadd
      3 ?        00:00:00 rcu_gp
      4 ?        00:00:00 rcu_par_gp
      6 ?        00:00:00 kworker/0:0H-kblockd
      7 ?        00:00:00 kworker/u30:0-events_unbound
      8 ?        00:00:00 mm_percpu_wq
      9 ?        00:00:00 ksoftirqd/0
     10 ?        00:00:00 rcu_sched
     11 ?        00:00:00 migration/0
     12 ?        00:00:00 watchdog/0
     13 ?        00:00:00 cpuhp/0
     15 ?        00:00:00 kdevtmpfs
     16 ?        00:00:00 netns
     17 ?        00:00:00 kauditd
     18 ?        00:00:00 xenbus
```

```
[ec2-user@samplegoogle ~]$ ps -Af
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 04:11 ?        00:00:02 /usr/lib/systemd/systemd --switched-root --system --deserialize 16
root           2       0  0 04:11 ?        00:00:00 [kthreadd]
root           3       2  0 04:11 ?        00:00:00 [rcu_gp]
root           4       2  0 04:11 ?        00:00:00 [rcu_par_gp]
root           6       2  0 04:11 ?        00:00:00 [kworker/0:0H-kblockd]
root           7       2  0 04:11 ?        00:00:00 [kworker/u30:0-events_unbound]
root           8       2  0 04:11 ?        00:00:00 [mm_percpu_wq]
root           9       2  0 04:11 ?        00:00:00 [ksoftirqd/0]
root          10       2  0 04:11 ?        00:00:00 [rcu_sched]
root          11       2  0 04:11 ?        00:00:00 [migration/0]
root          12       2  0 04:11 ?        00:00:00 [watchdog/0]
root          13       2  0 04:11 ?        00:00:00 [cpuhp/0]
root          15       2  0 04:11 ?        00:00:00 [kdevtmpfs]
root          16       2  0 04:11 ?        00:00:00 [netns]
root          17       2  0 04:11 ?        00:00:00 [kauditd]
root          18       2  0 04:11 ?        00:00:00 [xenbus]
root          19       2  0 04:11 ?        00:00:00 [xenwatch]
root          21       2  0 04:11 ?        00:00:00 [khungtaskd]
```

ps ux :

(OR) ----------It display all processas owned by a specific user.

```
[ec2-user@samplegoogle ~]$ ps ux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
ec2-user    1273  0.0  1.1  93736  9508 ?        Ss   04:12   0:00 /usr/lib/systemd/systemd --user
ec2-user    1278  0.0  0.5 170988  4604 ?        S    04:12   0:00 (sd-pam)
ec2-user    1284  0.0  0.7 161516  6416 ?        S    04:12   0:00 sshd: ec2-user@pts/0
ec2-user    1286  0.0  0.6 163256  5252 ?        S    04:12   0:00 sshd: ec2-user@notty
ec2-user    1287  0.0  0.4  23256  3876 pts/0    Ss+  04:12   0:00 -bash
ec2-user    1310  0.0  0.6  45848  5256 ?        Ss   04:12   0:00 /usr/libexec/openssh/sftp-server
ec2-user    1334  0.0  0.7 161528  6564 ?        S    04:18   0:00 sshd: ec2-user@pts/1
ec2-user    1336  0.0  0.6 163256  5260 ?        S    04:18   0:00 sshd: ec2-user@notty
ec2-user    1337  0.0  0.4  23256  3808 pts/1    Ss   04:18   0:00 -bash
ec2-user    1360  0.0  0.5  37468  4784 ?        Ss   04:18   0:00 /usr/libexec/openssh/sftp-server
ec2-user    1393  0.0  0.4  57820  3904 pts/1    R+   04:26   0:00 ps ux
[ec2-user@samplegoogle ~]$
```

ps U ec2-user:

```
[ec2-user@samplegoogle ~]$ ps U ec2-user
    PID TTY      STAT   TIME COMMAND
   1273 ?        Ss     0:00 /usr/lib/systemd/systemd --user
   1278 ?        S      0:00 (sd-pam)
   1284 ?        S      0:00 sshd: ec2-user@pts/0
   1286 ?        S      0:00 sshd: ec2-user@notty
   1287 pts/0    Ss+    0:00 -bash
   1310 ?        Ss     0:00 /usr/libexec/openssh/sftp-server
   1334 ?        S      0:00 sshd: ec2-user@pts/1
   1336 ?        S      0:00 sshd: ec2-user@notty
   1337 pts/1    Ss     0:00 -bash
   1360 ?        Ss     0:00 /usr/libexec/openssh/sftp-server
   1397 pts/1    R+     0:00 ps U ec2-user
[ec2-user@samplegoogle ~]$
```

ps aux : It will display every process on the system.

```
[ec2-user@samplegoogle ~]$ ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.2  1.6 179160 13476 ?        Ss   04:11   0:02 /usr/lib/systemd/systemd --switched-root --system --des
root           2  0.0  0.0      0     0 ?        S    04:11   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   04:11   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   04:11   0:00 [rcu_par_gp]
root           6  0.0  0.0      0     0 ?        I<   04:11   0:00 [kworker/0:0H-kblockd]
root           7  0.0  0.0      0     0 ?        I    04:11   0:00 [kworker/u30:0-events_unbound]
root           8  0.0  0.0      0     0 ?        I<   04:11   0:00 [mm_percpu_wq]
root           9  0.0  0.0      0     0 ?        S    04:11   0:00 [ksoftirqd/0]
root          10  0.0  0.0      0     0 ?        I    04:11   0:00 [rcu_sched]
root          11  0.0  0.0      0     0 ?        S    04:11   0:00 [migration/0]
root          12  0.0  0.0      0     0 ?        S    04:11   0:00 [watchdog/0]
root          13  0.0  0.0      0     0 ?        S    04:11   0:00 [cpuhp/0]
root          15  0.0  0.0      0     0 ?        S    04:11   0:00 [kdevtmpfs]
root          16  0.0  0.0      0     0 ?        I<   04:11   0:00 [netns]
root          17  0.0  0.0      0     0 ?        S    04:11   0:00 [kauditd]
root          18  0.0  0.0      0     0 ?        S    04:11   0:00 [xenbus]
root          19  0.0  0.0      0     0 ?        S    04:11   0:00 [xenwatch]
root          21  0.0  0.0      0     0 ?        S    04:11   0:00 [khungtaskd]
root          22  0.0  0.0      0     0 ?        S    04:11   0:00 [oom_reaper]
root          23  0.0  0.0      0     0 ?        I<   04:11   0:00 [writeback]
root          24  0.0  0.0      0     0 ?        S    04:11   0:00 [kcompactd0]
root          25  0.0  0.0      0     0 ?        SN   04:11   0:00 [ksmd]
root          26  0.0  0.0      0     0 ?        SN   04:11   0:00 [khugepaged]
root          27  0.0  0.0      0     0 ?        I<   04:11   0:00 [crypto]
```

pstree : It will display a tree of processes.

```
[ec2-user@samplegoogle ~]$ pstree
systemd─┬─NetworkManager───2*[{NetworkManager}]
        ├─2*[agetty]
        ├─auditd───{auditd}
        ├─chronyd
        ├─crond
        ├─dbus-daemon───{dbus-daemon}
        ├─polkitd───5*[{polkitd}]
        ├─rngd───{rngd}
        ├─rsyslogd───2*[{rsyslogd}]
        ├─sshd─┬─sshd───sshd───bash
        │      ├─2*[sshd───sshd───sftp-server]
        │      └─sshd───sshd───bash───pstree
        ├─sssd─┬─sssd_be
        │      └─sssd_nss
        ├─systemd───(sd-pam)
        ├─systemd-journal
        ├─systemd-logind
        ├─systemd-udevd
        └─tuned───3*[{tuned}]
[ec2-user@samplegoogle ~]$
```

pstree -A :

-A ---> Use ASCII characters to draw the tree

```
[ec2-user@samplegoogle ~]$ pstree -A
systemd-+-NetworkManager---2*[{NetworkManager}]
        |-2*[agetty]
        |-auditd---{auditd}
        |-chronyd
        |-crond
        |-dbus-daemon---{dbus-daemon}
        |-polkitd---5*[{polkitd}]
        |-rngd---{rngd}
        |-rsyslogd---2*[{rsyslogd}]
        |-sshd-+-sshd---sshd---bash
        |      |-2*[sshd---sshd---sftp-server]
        |      `-sshd---sshd---bash---pstree
        |-sssd-+-sssd_be
        |      `-sssd_nss
        |-systemd---(sd-pam)
        |-systemd-journal
        |-systemd-logind
        |-systemd-udevd
        `-tuned---3*[{tuned}]
[ec2-user@samplegoogle ~]$
```

pstree -G :

-G ---> Use VT100 line drawing characters.

```
[ec2-user@samplegoogle ~]$ pstree -G
systemd─┬─NetworkManager───2*[{NetworkManager}]
        ├─2*[agetty]
        ├─auditd───{auditd}
        ├─chronyd
        ├─crond
        ├─dbus-daemon───{dbus-daemon}
        ├─polkitd───5*[{polkitd}]
        ├─rngd───{rngd}
        ├─rsyslogd───2*[{rsyslogd}]
        ├─sshd─┬─sshd───sshd───bash
        │      ├─2*[sshd───sshd───sftp-server]
        │      └─sshd───sshd───bash───pstree
        ├─sssd─┬─sssd_be
        │      └─sssd_nss
        ├─systemd───(sd-pam)
        ├─systemd-journal
        ├─systemd-logind
        ├─systemd-udevd
        └─tuned───3*[{tuned}]
[ec2-user@samplegoogle ~]$
```

Note:

-A ---> This opion uses ASCII characters to draw the tree.

-G --->This opion uses VT100 line drawing characters.

```
[ec2-user@samplegoogle ~]$ ps -eg | grep crond
   1426 pts/1    S+     0:00 grep --color=auto crond LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=
40;33;01:cd=40;33;01:or=40;31;01:mi=01;05;37;41:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31
:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:
*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.zst=01
;31:*.tzst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;3
1:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*
.wim=01;31:*.swm=01;31:*.dwm=01;31:*.esd=01;31:*.jpg=01;35:*.jpeg=01;35:*.mjpg=01;35:*.mjpeg=01;35:*.gif=01;35:*.bmp=01;35
:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:
*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.ogm=01;35
:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*
.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf
=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=01;36:*.au=01;36:*.flac=01;36:*.m4a=01;36:*.mid=01;36:*.midi=01;36:*.mka=01;36:*.mp3=
01;36:*.mpc=01;36:*.ogg=01;36:*.ra=01;36:*.wav=01;36:*.oga=01;36:*.opus=01;36:*.spx=01;36:*.xspf=01;36: SSH_CONNECTION=157
.45.2.130 60343 172.31.31.192 22 LANG=en_US.UTF-8 HISTCONTROL=ignoredups HOSTNAME=samplegoogle.com XDG_SESSION_ID=4 USER=e
c2-user SELINUX_ROLE_REQUESTED= PWD=/home/ec2-user HOME=/home/ec2-user SSH_CLIENT=157.45.2.130 60343 22 SELINUX_LEVEL_REQU
ESTED= SSH_TTY=/dev/pts/1 MAIL=/var/spool/mail/ec2-user TERM=xterm SHELL=/bin/bash SELINUX_USE_CURRENT_RANGE= SHLVL=1 LOGN
AME=ec2-user DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus XDG_RUNTIME_DIR=/run/user/1000 PATH=/home/ec2-user/.loc
al/bin:/home/ec2-user/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin HISTSIZE=1000 LESSOPEN=||/usr/bin/lesspipe.sh
%s _=/usr/bin/grep
[ec2-user@samplegoogle ~]$
```

#ps -eg | grep crond

#ps -ef | grep crond

```
[ec2-user@samplegoogle ~]$ ps -eg | grep crond
   1426 pts/1    S+     0:00 grep --color=auto crond LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=
40;33;01:cd=40;33;01:or=40;31;01:mi=01;05;37;41:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31
:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:
*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.zst=01
;31:*.tzst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;3
1:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*
.wim=01;31:*.swm=01;31:*.dwm=01;31:*.esd=01;31:*.jpg=01;35:*.jpeg=01;35:*.mjpg=01;35:*.mjpeg=01;35:*.gif=01;35:*.bmp=01;35
:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:
*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.ogm=01;35
:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*
.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf
=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=01;36:*.au=01;36:*.flac=01;36:*.m4a=01;36:*.mid=01;36:*.midi=01;36:*.mka=01;36:*.mp3=
01;36:*.mpc=01;36:*.ogg=01;36:*.ra=01;36:*.wav=01;36:*.oga=01;36:*.opus=01;36:*.spx=01;36:*.xspf=01;36: SSH_CONNECTION=157
.45.2.130 60343 172.31.31.192 22 LANG=en_US.UTF-8 HISTCONTROL=ignoredups HOSTNAME=samplegoogle.com XDG_SESSION_ID=4 USER=e
c2-user SELINUX_ROLE_REQUESTED= PWD=/home/ec2-user HOME=/home/ec2-user SSH_CLIENT=157.45.2.130 60343 22 SELINUX_LEVEL_REQU
ESTED= SSH_TTY=/dev/pts/1 MAIL=/var/spool/mail/ec2-user TERM=xterm SHELL=/bin/bash SELINUX_USE_CURRENT_RANGE= SHLVL=1 LOGN
AME=ec2-user DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus XDG_RUNTIME_DIR=/run/user/1000 PATH=/home/ec2-user/.loc
al/bin:/home/ec2-user/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin HISTSIZE=1000 LESSOPEN=||/usr/bin/lesspipe.sh
%s _=/usr/bin/grep
[ec2-user@samplegoogle ~]$ ps -ef | grep crond
root         957       1  0 04:11 ?        00:00:00 /usr/sbin/crond -n
ec2-user    1428    1337  0 04:38 pts/1    00:00:00 grep --color=auto crond
[ec2-user@samplegoogle ~]$
```

Signals in Linux

- Signals are a way of sending simple messages to processes. Most of these messages are already defined and can be found in <linux/signal.h>. However, signals can only be processed when the process is in user mode. If a signal has been sent to a process that is in kernel mode, it is dealt with immediately on returning to user mode.
- Every signal has a unique signal name, an abbreviation that begins with SIG (SIGINT for interrupt signal, for example). Each signal name is a macro which stands for a positive integer - the signal number for that kind of signal. Your programs should never make assumptions about the numeric code for a particular kind of signal, but rather refer to them always by the names defined. This is because the number for a given kind of signal can vary from system to system, but the meanings of the names are standardized and fairly uniform.
- Signals can be generated by the process itself, or they can be sent from one process to another. A variety of signals can be generated or delivered, and they have many uses for

programmers. (To see a complete list of signals in the Linux® environment, uses the command kill -l.)

kill: The command will sends the specified signal to the specified process

#kill -l: It will display all signal names. These are found in /usr/include/linux/signal.h

```
[ec2-user@samplegoogle ~]$ kill -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL       5) SIGTRAP
 6) SIGABRT      7) SIGBUS       8) SIGFPE       9) SIGKILL     10) SIGUSR1
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO       30) SIGPWR
31) SIGSYS      34) SIGRTMIN    35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3
38) SIGRTMIN+4  39) SIGRTMIN+5  40) SIGRTMIN+6  41) SIGRTMIN+7  42) SIGRTMIN+8
43) SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
[ec2-user@samplegoogle ~]$
```

Few Important Signals with its descriptions:

Signal	Value	Action	Comment
SIGHUP	1	Term	Hangup detected on controlling terminal or death of controlling process
SIGINT	2	Term	Interrupt from keyboard
SIGQUIT	3	Core	Quit from keyboard
SIGILL	4	Core	Illegal Instruction
SIGABRT	6	Core	Abort signal from abort(3)
SIGFPE	8	Core	Floating point exception
SIGKILL	9	Term	Kill signal
SIGSEGV	11	Core	Invalid memory reference
SIGPIPE	13	Term	Broken pipe: write to pipe with no readers
SIGALRM	14	Term	Timer signal from alarm(2)
SIGTERM	15	Term	Termination signal
SIGUSR1	30,10,16	Term	User-defined signal 1
SIGUSR2	31,12,17	Term	User-defined signal 2
SIGCHLD	20,17,18	Ign	Child stopped or terminated
SIGCONT	19,18,25	Cont	Continue if stopped
SIGSTOP	17,19,23	Stop	Stop process
SIGTSTP	18,20,24	Stop	Stop typed at tty
SIGTTIN	21,21,26	Stop	tty input for background process
SIGTTOU	22,22,27	Stop	tty output for background process

The most common signals used are

- 1 for reloading the process
- 3 for quit the process from keyboard.
- 9 for killing the process
- 15 for Terminating the process
- 20 for stopping the process

To kill the signal completely

- To kill the signal
- First find out the process running in the system, let's say by a user

#ps –u <user name>
#ps –u ec2-user
#kill <signal no><process id>
#kill -9 1337

```
[ec2-user@samplegoogle ~]$ ps -u ec2-user
    PID TTY          TIME CMD
   1273 ?        00:00:00 systemd
   1278 ?        00:00:00 (sd-pam)
   1284 ?        00:00:00 sshd
   1286 ?        00:00:00 sshd
   1287 pts/0    00:00:00 bash
   1310 ?        00:00:00 sftp-server
   1334 ?        00:00:00 sshd
   1336 ?        00:00:00 sshd
   1337 pts/1    00:00:00 bash
   1360 ?        00:00:00 sftp-server
   1455 ?        00:00:00 sshd
   1458 ?        00:00:00 sshd
   1459 pts/2    00:00:00 bash
   1482 ?        00:00:00 sftp-server
   1502 ?        00:00:00 sshd
   1504 pts/3    00:00:00 bash
   1528 ?        00:00:00 sshd
   1529 ?        00:00:00 sftp-server
   1542 pts/3    00:00:00 ps
[ec2-user@samplegoogle ~]$ kill -9 1337
[ec2-user@samplegoogle ~]$ ps -u ec2-user
    PID TTY          TIME CMD
   1273 ?        00:00:00 systemd
   1278 ?        00:00:00 (sd-pam)
   1284 ?        00:00:00 sshd
   1286 ?        00:00:00 sshd
   1287 pts/0    00:00:00 bash
   1310 ?        00:00:00 sftp-server
   1334 ?        00:00:00 sshd
   1336 ?        00:00:00 sshd
   1360 ?        00:00:00 sftp-server
   1455 ?        00:00:00 sshd
   1458 ?        00:00:00 sshd
   1459 pts/2    00:00:00 bash
   1482 ?        00:00:00 sftp-server
   1502 ?        00:00:00 sshd
   1504 pts/3    00:00:00 bash
   1528 ?        00:00:00 sshd
   1529 ?        00:00:00 sftp-server
   1543 pts/3    00:00:00 ps
[ec2-user@samplegoogle ~]$
```

Monitoring the process using top command

- When you need to see the running processes on your Linux in real time, you have top as your tool for that.
- top also displays other info besides the running processes, like free memory both physical and swap.

Monitoring all process using top command.

- To monitor all processes in the system use the following command

#top

```
[ec2-user@samplegoogle ~]$ top
top - 04:57:36 up 46 min,  3 users,  load average: 0.00, 0.00, 0.00
Tasks: 100 total,   2 running,  98 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :    816.0 total,    374.0 free,    163.8 used,    278.2 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.    517.1 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    694 chrony    20   0  128052   3620   3176 S   5.6   0.4   0:00.05 chronyd
      1 root      20   0  179160  13476   9148 S   0.0   1.6   0:02.63 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-kblockd
      7 root      20   0       0      0      0 I   0.0   0.0   0:00.00 kworker/u30:0-events_unbound
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
      9 root      20   0       0      0      0 S   0.0   0.0   0:00.01 ksoftirqd/0
     10 root      20   0       0      0      0 I   0.0   0.0   0:00.09 rcu_sched
     11 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 migration/0
     12 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 watchdog/0
     13 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
     15 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kdevtmpfs
     16 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
     17 root      20   0       0      0      0 S   0.0   0.0   0:00.02 kauditd
     18 root      20   0       0      0      0 S   0.0   0.0   0:00.00 xenbus
     19 root      20   0       0      0      0 S   0.0   0.0   0:00.00 xenwatch
     21 root      20   0       0      0      0 S   0.0   0.0   0:00.00 khungtaskd
     22 root      20   0       0      0      0 S   0.0   0.0   0:00.00 oom_reaper
     23 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 writeback
     24 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kcompactd0
     25 root      25   5       0      0      0 S   0.0   0.0   0:00.00 ksmd
     26 root      39  19       0      0      0 S   0.0   0.0   0:00.00 khugepaged
     27 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 crypto
[ec2-user@samplegoogle ~]$
```

The first line in top

```
top - 04:57:36 up 46 min,  3 users,  load average: 0.00, 0.00, 0.00
```

“04:57:36” is the current time; “up 46 mins” shows how long the system has been up for; “3 user” how many users are logged in; “load average: 0.00, 0.00, 0.00” the load average of the system.

The second line in top

```
Tasks: 100 total,   2 running,  98 sleeping,   0 stopped,   0 zombie
```

Shows the number of processes and their current state.

The Third line in top

```
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
```

Shows CPU utilization details.

The fourth line in top

```
MiB Mem :    816.0 total,    374.0 free,    163.8 used,    278.2 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.    517.1 avail Mem
```

“816.0 total” is total memory in the system; “163 used” is the part of the RAM that currently contains information; “374.0 free” is the part of RAM that contains no information; “278.2 buffers and 517.1 cached” is the buffered and cached data for IO.

By default, top starts by showing the following task's property:

```
    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
```

Field	Description
PID	Process ID
USER	Effective User ID
PR	Dynamic priority
NI	Nice value, also known as base priority
VIRT	Virtual Size of the task. This includes the size of process's executable binary, the data area and all the loaded shared libraries.
RES	The size of RAM currently consumed by the task. Swapped out portion of the task is not included.
SHR	Some memory areas could be shared between two or more task, this field reflects that shared areas. The example of shared area are shared library and SysV shared memory.
S	Task status
%CPU	The percentage of CPU time dedicated to run the task since the last top's screen update.
%MEM	The percentage of RAM currently consumed by the task.
TIME+	The total CPU time the task has been used since it started. "+" sign means it is displayed with hundredth of a second granularity. By default, TIME/TIME+ doesn't account the CPU time used by the task's dead children.
Command	Showing program names

Interacting with TOP

Now that we are able to understand the output from TOP lets learn how to change the way the output is displayed.

Just press the following key while running top and the output will be sorted in real time.

- M – Sort by memory usage
- P – Sort by CPU usage
- T – Sort by cumulative time
- z – Color display
- k – Kill a process
- q – quit
- r – to renice a process
- h - help

```
[ec2-user@samplegoogle ~]$ top
top - 05:25:22 up  1:13,  4 users,  load average: 0.00, 0.00, 0.00
Tasks: 104 total,   1 running, 103 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :    816.0 total,    368.3 free,    169.0 used,    278.7 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.    511.7 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
      1 root      20   0  179160  13480   9148 S   0.0   1.6   0:02.66 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-kblockd
      7 root      20   0       0      0      0 I   0.0   0.0   0:00.00 kworker/u30:0-events_unbound
```

Finding RAM details in Linux

free: To find the amount of free and used RAM memory in the system.

Syntax :
free [-b|-k|-m|-g] [-l] [-o] [-t] [-s delay] [-c count] [-V]
-b,-k,-m,-g show output in bytes, KB, MB, or GB
-l show detailed low and high memory statistics
-o use old format (no -/+buffers/cache line)
-t display total for RAM + swap
-s update every [delay] seconds
-c update [count] times
-V display version information and exit

#free
#free -b
#free -k
#free -m
#free -g

```
[ec2-user@samplegoogle ~]$ free
              total        used        free      shared  buff/cache   available
Mem:         835572      172612      377592       12428      285368      524476
Swap:             0           0           0
[ec2-user@samplegoogle ~]$ free -b
              total        used        free      shared  buff/cache   available
Mem:      855625728   176791552   386592768    12726272   292241408   537026560
Swap:             0           0           0
[ec2-user@samplegoogle ~]$ free -k
              total        used        free      shared  buff/cache   available
Mem:         835572      172588      377592       12428      285392      524500
Swap:             0           0           0
[ec2-user@samplegoogle ~]$ free -m
              total        used        free      shared  buff/cache   available
Mem:            815         168         368          12         278         512
Swap:             0           0           0
[ec2-user@samplegoogle ~]$ free -g
              total        used        free      shared  buff/cache   available
Mem:              0           0           0           0           0           0
Swap:             0           0           0
```

#free –t
#free –V
#free –l

```
[ec2-user@samplegoogle ~]$ free -t
              total        used        free      shared  buff/cache   available
Mem:         835572      172648      377532       12428      285392      524440
Swap:             0           0           0
Total:       835572      172648      377532
[ec2-user@samplegoogle ~]$ free -V
free from procps-ng 3.3.15
[ec2-user@samplegoogle ~]$ free -l
              total        used        free      shared  buff/cache   available
Mem:         835572      172588      377592       12428      285392      524500
Low:         835572      457980      377592
High:             0           0           0
Swap:             0           0           0
[ec2-user@samplegoogle ~]$
```

File Achieve/Extraction Commands

zip: package and compress the files
Syntax:
zip <name of the zip file><list of the files>
Ex:
zip myzip devops.txt psddevops.txt

unzip: extract the files.
Syntax:
unzip <name of the zip file>
Ex:
unzip myzip.zip

```
[ec2-user@samplegoogle sample_folder]$ zip myzip devops.txt psddevops.txt
  adding: devops.txt (stored 0%)
  adding: psddevops.txt (stored 0%)
[ec2-user@samplegoogle sample_folder]$ ll
total 4
-rwxr-x--x. 1 ec2-user ec2-user   0 Oct 15 02:37 devops.txt
-rw-rw-r--. 1 ec2-user ec2-user 324 Oct 19 09:51 myzip.zip
-rwxrwxrwx. 1 ec2-user ec2-user   0 Oct 15 02:32 psddevops.txt
[ec2-user@samplegoogle sample_folder]$ rm devops.txt psddevops.txt
[ec2-user@samplegoogle sample_folder]$ ll
total 4
-rw-rw-r--. 1 ec2-user ec2-user 324 Oct 19 09:51 myzip.zip
[ec2-user@samplegoogle sample_folder]$ unzip myzip.zip
Archive:  myzip.zip
 extracting: devops.txt
 extracting: psddevops.txt
[ec2-user@samplegoogle sample_folder]$ ll
total 4
-rwxr-x--x. 1 ec2-user ec2-user   0 Oct 15 02:37 devops.txt
-rw-rw-r--. 1 ec2-user ec2-user 324 Oct 19 09:51 myzip.zip
-rwxrwxrwx. 1 ec2-user ec2-user   0 Oct 15 02:32 psddevops.txt
[ec2-user@samplegoogle sample_folder]$
```

tar: It is used to archive the directory/file.
Syntax: tar options directory/file The Options may be follows.

- c – Create a new archive
- v – Verbosely list files which are processed.
- f – Following is the archive file name

#tar -cvf mytar.tar devops.txt myzip.zip psddevops.txt
Creating an uncompressed tar archive using option cvf

tar -xvf mytar.tar
Extracting the files from achive.

tar -tvf mytar.tar
Viewing the files from archive.

```
[ec2-user@samplegoogle sample_folder]$ tar -tvf mytar.tar
-rwxr-x--x ec2-user/ec2-user 0 2020-10-15 02:37 devops.txt
-rw-rw-r-- ec2-user/ec2-user 324 2020-10-19 09:51 myzip.zip
-rwxrwxrwx ec2-user/ec2-user   0 2020-10-15 02:32 psddevops.txt
[ec2-user@samplegoogle sample_folder]$
```

```
[ec2-user@samplegoogle sample_folder]$ ll
total 4
-rwxr-x--x. 1 ec2-user ec2-user   0 Oct 15 02:37 devops.txt
-rw-rw-r--. 1 ec2-user ec2-user 324 Oct 19 09:51 myzip.zip
-rwxrwxrwx. 1 ec2-user ec2-user   0 Oct 15 02:32 psddevops.txt
[ec2-user@samplegoogle sample_folder]$ tar -cvf mytar.tar devops.txt myzip.zip psddevops.txt
devops.txt
myzip.zip
psddevops.txt
[ec2-user@samplegoogle sample_folder]$ ll
total 16
-rwxr-x--x. 1 ec2-user ec2-user     0 Oct 15 02:37 devops.txt
-rw-rw-r--. 1 ec2-user ec2-user 10240 Oct 19 10:06 mytar.tar
-rw-rw-r--. 1 ec2-user ec2-user   324 Oct 19 09:51 myzip.zip
-rwxrwxrwx. 1 ec2-user ec2-user     0 Oct 15 02:32 psddevops.txt
[ec2-user@samplegoogle sample_folder]$ rm devops.txt myzip.zip psddevops.txt
[ec2-user@samplegoogle sample_folder]$ ll
total 12
-rw-rw-r--. 1 ec2-user ec2-user 10240 Oct 19 10:06 mytar.tar
[ec2-user@samplegoogle sample_folder]$ tar -xvf mytar.tar
devops.txt
myzip.zip
psddevops.txt
[ec2-user@samplegoogle sample_folder]$ ll
total 16
-rwxr-x--x. 1 ec2-user ec2-user     0 Oct 15 02:37 devops.txt
-rw-rw-r--. 1 ec2-user ec2-user 10240 Oct 19 10:06 mytar.tar
-rw-rw-r--. 1 ec2-user ec2-user   324 Oct 19 09:51 myzip.zip
-rwxrwxr-x. 1 ec2-user ec2-user     0 Oct 15 02:32 psddevops.txt
[ec2-user@samplegoogle sample_folder]$
```

#tar -cvzf mytar.tar.gz .
Creating a tar gzipped archive using option cvzf

#tar -xvzf mytar.tar.gz
Extract a gzipped tar archive (*.tar.gz) using option xvzf

#tar -tvzf mytar.tar.gz
View the *.tar.gz file content without extracting using option tvzf

```
[ec2-user@samplegoogle sample_folder]$ ll
total 0
-rwxr-x--x. 1 ec2-user ec2-user 0 Oct 15 02:37 devops.txt
-rwxrwxr-x. 1 ec2-user ec2-user 0 Oct 15 02:32 psddevops.txt
[ec2-user@samplegoogle sample_folder]$ tar -cvzf mytar.tar.gz .
./
./devops.txt
./psddevops.txt
[ec2-user@samplegoogle sample_folder]$ ll
total 4
-rwxr-x--x. 1 ec2-user ec2-user   0 Oct 15 02:37 devops.txt
-rw-rw-r--. 1 ec2-user ec2-user 170 Oct 19 10:14 mytar.tar.gz
-rwxrwxr-x. 1 ec2-user ec2-user   0 Oct 15 02:32 psddevops.txt
[ec2-user@samplegoogle sample_folder]$ rm devops.txt psddevops.txt
[ec2-user@samplegoogle sample_folder]$ ll
total 4
-rw-rw-r--. 1 ec2-user ec2-user 170 Oct 19 10:14 mytar.tar.gz
[ec2-user@samplegoogle sample_folder]$ tar -xvzf mytar.tar.gz
./
./devops.txt
./psddevops.txt
[ec2-user@samplegoogle sample_folder]$ ll
total 4
-rwxr-x--x. 1 ec2-user ec2-user   0 Oct 15 02:37 devops.txt
-rw-rw-r--. 1 ec2-user ec2-user 170 Oct 19 10:14 mytar.tar.gz
-rwxrwxr-x. 1 ec2-user ec2-user   0 Oct 15 02:32 psddevops.txt
[ec2-user@samplegoogle sample_folder]$ tar -tvzf mytar.tar.gz
drwxrwxrwx ec2-user/ec2-user 0 2020-10-19 10:13 ./
-rwxr-x--x ec2-user/ec2-user 0 2020-10-15 02:37 ./devops.txt
-rwxrwxr-x ec2-user/ec2-user 0 2020-10-15 02:32 ./psddevops.txt
[ec2-user@samplegoogle sample_folder]$
```

#tar -cvzf mytar.tgz .
Creating a tar gzipped archive using option cvzf

#tar -xvzf mytar.tgz
Extract a gzipped tar archive (*.tar.gz) using option xvzf

#tar -tvzf mytar.tgz
View the *.tar.gz file content without extracting using option tvzf

```
[ec2-user@samplegoogle sample_folder]$ ll
total 0
-rwxr-x--x. 1 ec2-user ec2-user 0 Oct 15 02:37 devops.txt
-rwxrwxr-x. 1 ec2-user ec2-user 0 Oct 15 02:32 psddevops.txt
[ec2-user@samplegoogle sample_folder]$ tar -cvzf mytar.tgz .
./
./devops.txt
./psddevops.txt
[ec2-user@samplegoogle sample_folder]$ ll
total 4
-rwxr-x--x. 1 ec2-user ec2-user   0 Oct 15 02:37 devops.txt
-rw-rw-r--. 1 ec2-user ec2-user 170 Oct 19 10:20 mytar.tgz
-rwxrwxr-x. 1 ec2-user ec2-user   0 Oct 15 02:32 psddevops.txt
[ec2-user@samplegoogle sample_folder]$ rm devops.txt psddevops.txt
[ec2-user@samplegoogle sample_folder]$ ll
total 4
-rw-rw-r--. 1 ec2-user ec2-user 170 Oct 19 10:20 mytar.tgz
[ec2-user@samplegoogle sample_folder]$ tar -xvzf mytar.tgz
./
./devops.txt
./psddevops.txt
[ec2-user@samplegoogle sample_folder]$ ll
total 4
-rwxr-x--x. 1 ec2-user ec2-user   0 Oct 15 02:37 devops.txt
-rw-rw-r--. 1 ec2-user ec2-user 170 Oct 19 10:20 mytar.tgz
-rwxrwxr-x. 1 ec2-user ec2-user   0 Oct 15 02:32 psddevops.txt
[ec2-user@samplegoogle sample_folder]$ tar -tvzf mytar.tgz
drwxrwxr-x ec2-user/ec2-user 0 2020-10-19 10:19 ./
-rwxr-x--x ec2-user/ec2-user 0 2020-10-15 02:37 ./devops.txt
-rwxrwxr-x ec2-user/ec2-user 0 2020-10-15 02:32 ./psddevops.txt
[ec2-user@samplegoogle sample_folder]$
```

USER AND GROUP ADMINISTRATION

In Linux/Unix user is one who uses the system. There can be at least one or more than one users in Linux at a time. Users on a system are identified by a username and a userid. The username is something that users would normally refer to, but as far as the operating system is concerned this is referred to using the user id (or uid). The username is typically a user friendly string, such as your name, whereas the user id is a number. The words username and userid are often (incorrectly) used interchangeably. The user id numbers should be unique (one number per user). If you had two usernames with the same user id, effectively there permissions would be the same and the files that they create would appear to have been created by the same user. This should not be allowed and the **useradd** command will not allow usernames to share the same userid.

Some Important Points related to Users:

- Users and groups are used to control access to files and resources.
- Users login to the system by supplying their username and password.
- Every file on the system is owned by a user and associated with a group.

- Every process has an owner and group affiliation, and can only access the resources its owner or group can access.
- Every user of the system is assigned a unique user ID number (the UID).
- Users name and UID are stored in **/etc/passwd.**
- User's password is stored in **/etc/shadow** in encrypted form.
- Users are assigned a **home directory** and a program that is run when they login (**Usually a shell**).
- Users cannot read, write or execute each other's files without permission.

Types of users In Linux and their attributes:

TYPE	EXAMPLE	USER ID (UID)	GROUP ID (GID)	HOME DIRECTORY	SHELL
Super User	Root	0	0	/root	/bin/bash
System User	ftp, ssh, apache nobody	1 to 499	1 to 499	/var/ftp , etc	/sbin/nologin
Normal User	Visitor, ktuser,etc	500 to 60000	500 to 60000	/home/user name	/bin/bash

In Linux there are three types of users

1. Super user or root user

Super user or the root user is the most powerful user. He is the administrator user.

2. System user

System users are the users created by the softwares or applications. For example if we install Apache it will create a user apache. These kinds of users are known as system users.

3. Normal user

Normal users are the users created by root user. They are normal users like Rahul, Musab etc. Only the root user has the permission to create or remove a user.

Whenever a user is created in Linux things created by default:-

- A home directory is created(/home/username)
- A mail box is created(/var/spool/mail)
- unique UID & GID are given to user

Linux uses UPG (User Private Group) scheme

- It means that whenever a user is created is has its own private group
- For Example if a user is created with the name Rahul, then a primary group for that user will be Rahul only

There are two important files a user administrator should be aware of.

1. "/etc/passwd"
2. "/etc/shadow"

Each of the above mentioned files have specific formats.

1. /etc/passwd

```
[ec2-user@samplegoogle ~]$ head /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
```

The above fields are

- root =name
- x= link to password file i.e. /etc/shadow
- 0 or 1= UID (user id)
- 0 or 1=GID (group id)
- root or bin = comment (brief information about the user)
- /root or /bin = home directory of the user
- /bin/bash or /sbin/nologin = shell

2. /etc/shadow

```
[root@samplegoogle ec2-user]# head /etc/shadow
root:!!:18375:0:99999:7:::
bin:*:18199:0:99999:7:::
```

The fields are as follows

1. root = User name
2. :$1fdsfsgsdfsdkffefje = Encrypted password
3. 18375= Days since that password was last changed.
4. 0 = Days after which password must be changed.
5. 99999 = Days before password is to expire that user is warned.
6. 7 = Days after the password is expires that the user is disabled.
7. A reserved field.

Password Complexity Requirements:

- A root user can change password of self and of any user in the system, there are no rules for root to assign a password. Root can assign any length of password either long or short, it can be alphabet or numeric or both. On the whole there is no limitation for root for assigning a password.
- A normal user can change only its password. Valid password for a normal user should adhere to the following rules.
- It should be at least 7 characters but not more than 255 characters.
- At least one character should be Upper case.
- At least one character should be Lower case.
- At least one character should be a symbol, and one character should be a number.
- It should not match the previous password.

- It cannot have a sequence (ex: 123456 or abcdef).
- The login name and the password cannot be same.

Note: For security reasons don't keep the password based on date of birth because it can easily be hacked.

1. Creating the user

```
[root@samplegoogle ec2-user]# useradd --help
Usage: useradd [options] LOGIN
       useradd -D
       useradd -D [options]

Options:
  -b, --base-dir BASE_DIR       base directory for the home directory of the
                                new account
  -c, --comment COMMENT         GECOS field of the new account
  -d, --home-dir HOME_DIR       home directory of the new account
  -D, --defaults                print or change default useradd configuration
  -e, --expiredate EXPIRE_DATE  expiration date of the new account
  -f, --inactive INACTIVE       password inactivity period of the new account
  -g, --gid GROUP               name or ID of the primary group of the new
                                account
  -G, --groups GROUPS           list of supplementary groups of the new
                                account
  -h, --help                    display this help message and exit
  -k, --skel SKEL_DIR           use this alternative skeleton directory
  -K, --key KEY=VALUE           override /etc/login.defs defaults
  -l, --no-log-init             do not add the user to the lastlog and
                                faillog databases
  -m, --create-home             create the user's home directory
  -M, --no-create-home          do not create the user's home directory
  -N, --no-user-group           do not create a group with the same name as
                                the user
  -o, --non-unique              allow to create users with duplicate
                                (non-unique) UID
  -p, --password PASSWORD       encrypted password of the new account
  -r, --system                  create a system account
  -R, --root CHROOT_DIR         directory to chroot into
  -P, --prefix PREFIX_DIR       prefix directory where are located the /etc/* files
  -s, --shell SHELL             login shell of the new account
  -u, --uid UID                 user ID of the new account
  -U, --user-group              create a group with the same name as the user
  -Z, --selinux-user SEUSER     use a specific SEUSER for the SELinux user mapping
```

Syntax:

useradd <option><username>

Options are

- -u user id
- -G Secondary group id
- -g primary group id
- -d home directory
- -c comment
- -s shell

Create a user with default attributes

- When no option is used with **useradd** command the options like **UID, GID, home dir and shell** will be assigned default.
- #useradd <username>

#useradd devops

```
[root@samplegoogle ec2-user]# useradd devops
[root@samplegoogle ec2-user]# tail /etc/passwd
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
polkitd:x:998:996:User for polkitd:/:/sbin/nologin
unbound:x:997:995:Unbound DNS resolver:/etc/unbound:/sbin/nologin
sssd:x:996:993:User for sssd:/:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
chrony:x:995:992::/var/lib/chrony:/sbin/nologin
rngd:x:994:991:Random Number Generator Daemon:/var/lib/rngd:/sbin/nologin
ec2-user:x:1000:1000:Cloud User:/home/ec2-user:/bin/bash
ansadmin:x:1001:1001::/home/ansadmin:/bin/bash
devops:x:1002:1002::/home/devops:/bin/bash
[root@samplegoogle ec2-user]#
```

Observe that the **uid, gid, home dir, and shell** is assigned automatically.

Create a user with our own attributes

Create a user with following attributes.

- Name = devops1
- uid=505
- home dir = /home/myhome
- comment =devops purpose
- #useradd devops1 -u 505 -d /home/myhome -c devops_purpose

```
[root@samplegoogle ec2-user]# tail /etc/passwd
polkitd:x:998:996:User for polkitd:/:/sbin/nologin
unbound:x:997:995:Unbound DNS resolver:/etc/unbound:/sbin/nologin
sssd:x:996:993:User for sssd:/:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
chrony:x:995:992::/var/lib/chrony:/sbin/nologin
rngd:x:994:991:Random Number Generator Daemon:/var/lib/rngd:/sbin/nologin
ec2-user:x:1000:1000:Cloud User:/home/ec2-user:/bin/bash
ansadmin:x:1001:1001::/home/ansadmin:/bin/bash
devops:x:1002:1002::/home/devops:/bin/bash
devops1:x:505:1003:devops_purpose:/home/myhome:/bin/bash
[root@samplegoogle ec2-user]#
```

Assigning password to the user

- As a root user we can assign any password to any user
- The syntax for assigning a password is#passwd to assign password to current user (the one with which you have logged in, if it is root then root's password will be changed)
- #passwd <user name> to assign a password to a specific user, only root can assign password to other user.

```
[root@samplegoogle ec2-user]# passwd devops
Changing password for user devops.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@samplegoogle ec2-user]# su devops
[devops@samplegoogle ec2-user]$ cd
[devops@samplegoogle ~]$ pwd
/home/devops
[devops@samplegoogle ~]$
```

Modifying the user's attribute

- After creating a user if we need to modify the attributes of user like changing uid, changing secondary group id or adding a comment, locking or unlocking the user account, can be done by following command

Syntax

#usermod <options><username>

Options are:

All the options which are used with usermod command can be used and also the following,

- -l to change login name
- -L to LOCK account
- -U to UNLOCK account

Ex

usermod -l newname oldname (changing the name of the user)

Ex

usermod -L newname to lock the user account

Ex

usermod -U newname to unlock the user account

Note:

When an account is locked it will show! (Exclamation mark) in /etc/shadow file.

Locking User

#usermod -L devops1

```
[ec2-user@samplegoogle ~]$ sudo su
[root@samplegoogle ec2-user]# usermod -L devops1
[root@samplegoogle ec2-user]# tail /etc/shadow
polkitd:!!:18375::::::
unbound:!!:18375::::::
sssd:!!:18375::::::
sshd:!!:18375::::::
chrony:!!:18375::::::
rngd:!!:18375::::::
ec2-user:!!:18546:0:99999:7:::
ansadmin:$6$ILrT.UGCvoXCfUy6$gM0LilB1CbIP8Lao2QZF.D8njod7LXWGT1hzPC9sG2U88bhvpGPtiFQvTPLtH/a4RSGYTEiUm2NaZXmWa6frI/:18550:
0:99999:7:::
devops:$6$AR/fNYwmIQC.rMJZ$NW9PuPnFw2oDNDeHW0TjPvvoIcLl6TIJxZ4hq/atKkcvYnFfZiEundbp/ovk0stBKOGpAdkgRcns6Ndr5TaxU1:18554:0:
99999:7:::
devops1:!$6$DQlrtae7xskRcR28$.atZUAVuSM.r5Nx6e2EPcUVCaz5JQr8cBKf6/YPQpYjyfvkcjTLXFFOcNGyA6j2jpypnfekR34sUiFGAqhkT7/:18554:
0:99999:7:::
[root@samplegoogle ec2-user]#
```

Unlocking User

#usermod -U devops1

```
[root@samplegoogle ec2-user]# usermod -U devops1
[root@samplegoogle ec2-user]# tail /etc/shadow
polkitd:!!:18375::::::
unbound:!!:18375::::::
sssd:!!:18375::::::
sshd:!!:18375::::::
chrony:!!:18375::::::
rngd:!!:18375::::::
ec2-user:!!:18546:0:99999:7:::
ansadmin:$6$ILrT.UGCvoXCfUy6$gM0LilB1CbIP8Lao2QZF.D8njod7LXWGT1hzPC9sG2U88bhvpGPtiFQvTPLtH/a4RSGYTEiUm2NaZXmWa6frI/:18550:
0:99999:7:::
devops:$6$AR/fNYwmIQC.rMJZ$NW9PuPnFw2oDNDeHW0TjPvvoIcLl6TIJxZ4hq/atKkcvYnFfZiEundbp/ovk0stBKOGpAdkgRcns6Ndr5TaxU1:18554:0:
99999:7:::
devops1:$6$DQlrtae7xskRcR28$.atZUAVuSM.r5Nx6e2EPcUVCaz5JQr8cBKf6/YPQpYjyfvkcjTLXFFOcNGyA6j2jpypnfekR34sUiFGAqhkT7/:18554:0
:99999:7:::
[root@samplegoogle ec2-user]#
```

Renaming the user

#usermod -l aws devops1

```
[root@samplegoogle ec2-user]# usermod -l aws devops1
usermod: user devops1 is currently used by process 3301
[root@samplegoogle ec2-user]# kill -9 3301
[root@samplegoogle ec2-user]# usermod -l aws devops1
[root@samplegoogle ec2-user]# tail /etc/shadow
polkitd:!!:18375::::::
unbound:!!:18375::::::
sssd:!!:18375::::::
sshd:!!:18375::::::
chrony:!!:18375::::::
rngd:!!:18375::::::
ec2-user:!!:18546:0:99999:7:::
ansadmin:$6$ILrT.UGCvoXCfUy6$gM0LilB1CbIP8Lao2QZF.D8njod7LXWGT1hzPC9sG2U88bhvpGPtiFQvTPLtH/a4RSGYTEiUm2NaZXmWa6frI/:18550:
0:99999:7:::
devops:$6$AR/fNYwmIQC.rMJZ$NW9PuPnFw2oDNDeHW0TjPvvoIcLl6TIJxZ4hq/atKkcvYnFfZiEundbp/ovk0stBKOGpAdkgRcns6Ndr5TaxU1:18554:0:
99999:7:::
aws:$6$DQlrtae7xskRcR28$.atZUAVuSM.r5Nx6e2EPcUVCaz5JQr8cBKf6/YPQpYjyfvkcjTLXFFOcNGyA6j2jpypnfekR34sUiFGAqhkT7/:18554:0:999
99:7:::
[root@samplegoogle ec2-user]#
```

The password parameters

- For any user we can set the parameters for the password, like min and max password age, password expiration warnings and a/c expiration date etc.
- To view the advanced parameters of the user, use

#chage -l < user name>

#chage -l aws

```
[root@samplegoogle ec2-user]# chage -l aws
Last password change                                    : Oct 19, 2020
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
[root@samplegoogle ec2-user]#
```

- Last password change: When the password was change last time.
- Password expires: Password expiry date
- Password inactive: After password expiry grace period before the account gets locked.
- Account expires: Date on which the account expires.
- Minimum number of days b/w password change: once the password is changed, it cannot be changed until a min period of specified date. [0] means never.
- Max number of days b/w password change: After changing the password how long it will be valid for.
- Number of days of warning before password expires: start of warnings to change the password, no. of days before the password expires.

Changing the password parameters

Changing of the password parameters can be done by two ways.

1. #chage <user name >
2. #chage <option><value><username>

Let's see the first method and then the other.

- To set the password parameters of a user "aws" to
  - Min password age : 2 days
  - Max password age: 7 days
  - Password expiration warnings: 2 days before password expiresPassword inactive [-1]: 0 same day account is locked after password expiry.
  - A/C expiration date: 2020-12-31 (dec 31st 2020)

```
[root@samplegoogle ec2-user]# chage aws
Changing the aging information for aws
Enter the new value, or press ENTER for the default

        Minimum Password Age [0]: 2
        Maximum Password Age [99999]: 7
        Last Password Change (YYYY-MM-DD) [2020-10-19]:
        Password Expiration Warning [7]: 7
        Password Inactive [-1]: 0
        Account Expiration Date (YYYY-MM-DD) [-1]: 2020-12-31
[root@samplegoogle ec2-user]# chage -l aws
Last password change                                    : Oct 19, 2020
Password expires                                        : Oct 26, 2020
Password inactive                                       : Oct 26, 2020
Account expires                                         : Dec 31, 2020
Minimum number of days between password change          : 2
Maximum number of days between password change          : 7
Number of days of warning before password expires       : 7
[root@samplegoogle ec2-user]#
```

- The second method is for, if you want to change a particular field of password aging policy
- #chage <option><value><username>
- The options which can be used are as follows

-m for Min password age

-M for Max password age

-d for last time the password is changed.

-W Password expiration warnings

-I Password inactive [-1 means inactive].

-E A/C expiration date

```
[root@samplegoogle ec2-user]# chage -E 2021-01-30 aws
[root@samplegoogle ec2-user]# chage -l aws
Last password change                                    : Oct 19, 2020
Password expires                                        : Oct 26, 2020
Password inactive                                       : Oct 26, 2020
Account expires                                         : Jan 30, 2021
Minimum number of days between password change          : 2
Maximum number of days between password change          : 7
Number of days of warning before password expires       : 7
[root@samplegoogle ec2-user]#
```

Deleting a User:

- To delete a user the syntax used is

#userdel <username> it will only delete the user but home directory will be there. To delete the user with its home directory use the following command.

- #userdel –r < user name >
- #userdel devops
- #userdel –r aws

```
[root@samplegoogle ec2-user]# clear
[root@samplegoogle ec2-user]# userdel aws
[root@samplegoogle ec2-user]# tree /home
/home
├── ansadmin
├── devops
├── ec2-user
│   ├── 700
│   ├── devops_batch
│   │   ├── output.txt
│   │   ├── sample.txt
│   │   ├── sedfile.txt
│   │   └── temp.txt
```

```
[root@samplegoogle ec2-user]# userdel -r devops
[root@samplegoogle ec2-user]# tree /home
/home
├── ansadmin
├── ec2-user
│   ├── 700
│   ├── devops_batch
│   │   ├── output.txt
│   │   ├── sample.txt
│   │   ├── sedfile.txt
│   │   └── temp.txt
│   ├── devops.txt
│   ├── dir
│   │   └── dir1
│   │       └── dir2
│   │           └── dir3
```

GROUP ADMINISTRATION

GROUPS

- Users are assigned to groups with unique group ID numbers (the GID).
- The group name and GID are stored in /etc/group
- Each user is given their own private group
- They can also be added to their groups to gain additional access
- All users in a group can share files that belong to the group

Each user is a member of at least one group, called a primary group. In addition, a user can be a member of an unlimited number of secondary groups. Group membership can be used to control the files that a user can read and edit. For example, if two users are working on the same project you might put them in the same group so they can edit a particular file that other users cannot access.

- A user's primary group is defined in the /etc/passwd file and Secondary groups are defined in the /etc/group file.
- The primary group is important because files created by this user will inherit that group affiliation.

Creating a Group with default options:

To create a group the syntax is

#groupadd <name for the group>

#groupadd devopsgroup

```
[root@samplegoogle ec2-user]# groupadd devopsgroup
[root@samplegoogle ec2-user]# tai /etc/group
bash: tai: command not found
[root@samplegoogle ec2-user]# tail /etc/group
ssh_keys:x:994:
sssd:x:993:
sshd:x:74:
chrony:x:992:
rngd:x:991:
ec2-user:x:1000:
slocate:x:21:
ansadmin:x:1001:
devops1:x:1003:
devopsgroup:x:1004:
[root@samplegoogle ec2-user]#
```

Creating a group with user specified group id (GID)

#groupadd <option><name for the group>

#groupadd -g 600devopsgroup

```
[root@samplegoogle ec2-user]# groupadd -g 600 devopsgroup
[root@samplegoogle ec2-user]# tail /etc/group
ssh_keys:x:994:
sssd:x:993:
sshd:x:74:
chrony:x:992:
rngd:x:991:
ec2-user:x:1000:
slocate:x:21:
ansadmin:x:1001:
devops1:x:1003:
devopsgroup:x:600:
[root@samplegoogle ec2-user]#
```

Modifying the properties of the group

- To modify the group properties the syntax is

#groupmod <option><arguments><group name>

The options are

-g to change the group id

-o to override the previous assigned id, if it matches with the new one.

-n to change the group name

Changing the GID of the group

#groupmod –g 700 devopsgroup

Verify it in /etc/group

```
[root@samplegoogle ec2-user]# groupmod -g 700 devopsgroup
[root@samplegoogle ec2-user]# tail /etc/group
ssh_keys:x:994:
sssd:x:993:
sshd:x:74:
chrony:x:992:
rngd:x:991:
ec2-user:x:1000:
slocate:x:21:
ansadmin:x:1001:
devops1:x:1003:
devopsgroup:x:700:
[root@samplegoogle ec2-user]#
```

#groupmod -n awsgroup devopsgroup

```
[root@samplegoogle ec2-user]# groupmod -n awsgroup devopsgroup
[root@samplegoogle ec2-user]# tail /etc/group
ssh_keys:x:994:
sssd:x:993:
sshd:x:74:
chrony:x:992:
rngd:x:991:
ec2-user:x:1000:
slocate:x:21:
ansadmin:x:1001:
devops1:x:1003:
awsgroup:x:700:
[root@samplegoogle ec2-user]#
```

Adding and Removing Members to a Group

- Adding the members to the group is to add users to the group. To add the members to the group the syntaxes are

To add single user to the group

#usermod –G <group name >< user name>

#usermod -G awsgroup awsuser1

```
[root@samplegoogle ec2-user]# usermod -G awsgroup awsuser1
[root@samplegoogle ec2-user]# tail /etc/group
rngd:x:991:
ec2-user:x:1000:
slocate:x:21:
ansadmin:x:1001:
devops1:x:1003:
awsgroup:x:700:awsuser1
awsuser1:x:1002:
awsuser2:x:1004:
awsuser3:x:1005:
awsadmin:x:1006:
[root@samplegoogle ec2-user]#
```

Adding multiple single or multiple users to the group with various attributes

#gpasswd < option><arguments><group name>

Options:

-M For Adding Multiple users to a group
-A for Adding a group Administrator
-a for Adding a single user to a group
-d removing a user from a group

gpasswd -M awsuser2,awsuser3 awsgroup

```
[root@samplegoogle ec2-user]# gpasswd -M awsuser2,awsuser3 awsgroup
[root@samplegoogle ec2-user]# tail /etc/group
rngd:x:991:
ec2-user:x:1000:
slocate:x:21:
ansadmin:x:1001:
devops1:x:1003:
awsgroup:x:700:awsuser2,awsuser3
awsuser1:x:1002:
awsuser2:x:1004:
awsuser3:x:1005:
awsadmin:x:1006:
[root@samplegoogle ec2-user]#
```

#gpasswd -a awsuser1 awsgroup

```
[root@samplegoogle ec2-user]# gpasswd -a awsuser1 awsgroup
Adding user awsuser1 to group awsgroup
[root@samplegoogle ec2-user]# tail /etc/group
rngd:x:991:
ec2-user:x:1000:
slocate:x:21:
ansadmin:x:1001:
devops1:x:1003:
awsgroup:x:700:awsuser2,awsuser3,awsuser1
awsuser1:x:1002:
awsuser2:x:1004:
awsuser3:x:1005:
awsadmin:x:1006:
[root@samplegoogle ec2-user]#
```

#gpasswd -A awsadmin awsgroup
#tail /etc/gshadow

```
[root@samplegoogle ec2-user]# gpasswd -A awsadmin awsgroup
[root@samplegoogle ec2-user]# tail /etc/group
rngd:x:991:
ec2-user:x:1000:
slocate:x:21:
ansadmin:x:1001:
devops1:x:1003:
awsgroup:x:700:awsuser2,awsuser3,awsuser1,awsadmin
awsuser1:x:1002:
awsuser2:x:1004:
awsuser3:x:1005:
awsadmin:x:1006:
[root@samplegoogle ec2-user]# tail /etc/gshadow
rngd:!::
ec2-user:!::
slocate:!::
ansadmin:!::
devops1:!::
awsgroup:!:awsadmin:awsuser2,awsuser3,awsuser1,awsadmin
awsuser1:!::
awsuser2:!::
awsuser3:!::
awsadmin:!::
[root@samplegoogle ec2-user]#
```

#gpasswd -d awsuser3 awsgroup

#tail /etc/gshadow

```
[root@samplegoogle ec2-user]# gpasswd -d awsuser3 awsgroup
Removing user awsuser3 from group awsgroup
[root@samplegoogle ec2-user]# tail /etc/gshadow
rngd:!::
ec2-user:!::
slocate:!::
ansadmin:!::
devops1:!::
awsgroup:!:awsadmin:awsuser2,awsuser1,awsadmin
awsuser1:!::
awsuser2:!::
awsuser3:!::
awsadmin:!::
[root@samplegoogle ec2-user]#
```

Deleting a group

groupdel awsgroup

tail /etc/gshadow

```
[root@samplegoogle ec2-user]# groupdel  awsgroup
[root@samplegoogle ec2-user]# tail /etc/gshadow
chrony:!::
rngd:!::
ec2-user:!::
slocate:!::
ansadmin:!::
devops1:!::
awsuser1:!::
awsuser2:!::
awsuser3:!::
awsadmin:!::
[root@samplegoogle ec2-user]#
```

JOB AUTOMATION

Automation with cron and at

- In any operating system, it is possible to create jobs that you want to reoccur. This process, known as job scheduling, is usually done based on user-defined jobs. For Red Hat or any other Linux, this process is handled by the cron service or a daemon called crond, which can be used to schedule tasks (also called jobs). By default, Red Hat comes with a set of predefined jobs that occur on the system (hourly, daily, weekly, monthly, and with arbitrary periodicity). As an administrator, however, you can define your own jobs and allow your users to create them as well.
- The importance of the job scheduling is that the critical tasks like taking backups, which the clients usually wants to be taken in nights, can easily be performed without the intervention of the administrator by scheduling a cron job. If the cron job is scheduled carefully than the backup will be taken at any given time of the client and there will be no need for the administrator to remain back at nights to take the backup.

Important Files related to cron and at

- /etc/crontab is the file which stores all scheduled jobs.
- /etc/cron.deny is the file used to restrict the users from using cron jobs.
- /etc/cron.allow is used to allow only users whose names are mentioned in this file to use cron jobs. (this file does not exist by default).
- /etc/at.deny same as cron.deny for restricting at jobs.
- /etc/at.allow same as cron.allow for allowing user to use at jobs.

Crontab format

- To assign a job in the Crontab file the format used is the following.

```
# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name  command to be executed
```

Options	Explanation
*	Is treated as a wild card. Meaning any possible value.
*/5	Is treated as ever 5 minutes, hours, days, or months. Replacing the 5 with another numerical value will change this option.
2,4,6	Treated as an OR, so if placed in the hours, this could mean at 2, 4, or 6 o-clock.
9-17	Treats for any value between 9 and 17. So if placed in day of month this would be days 9 through 17. Or if put in hours it would be between 9 and 5.

Crontab Commands

Command	Explanation
crontab –e	Edit your crontab file, or create one if it doesn't already exist.
crontab –l	Display your crontab file.
crontab –r	Remove your crontab file.
crontab -u	If combined with **–e,** edit a particular user's Crontab file and if combined with **–l,** display a particular user's crontab file. If combined with **–r,** deletes a particular user's Crontab file

crontab –l
crontab –e
crontab –l
sudo service crond restart

```
[ec2-user@samplegoogle ~]$ crontab -l
no crontab for ec2-user
[ec2-user@samplegoogle ~]$ crontab -e
no crontab for ec2-user - using an empty one
crontab: installing new crontab
[ec2-user@samplegoogle ~]$ crontab -l
*/2 * * * * /home/ec2-user/sample_script.sh
[ec2-user@samplegoogle ~]$ sudo service crond restart
Redirecting to /bin/systemctl restart crond.service
[ec2-user@samplegoogle ~]$
```

Note: All logs will be stored in the "/var/log/cron" path

Disable the cron jobs

#crontab –e
#*/2 * * * * /home/ec2-user/sample_script.sh

```
#*/2 * * * * /home/ec2-user/sample_script.sh
~
~
~
```

Downloading files from Internet

curl:

curl is a command line tool to transfer data to or from a server, using any of the supported protocols (HTTP, FTP, IMAP, POP3, SCP, SFTP, SMTP, TFTP, TELNET, LDAP or FILE). curl is powered by Libcurl. This tool is preferred for automation, since it is designed to work without user interaction. curl can transfer multiple file at once

Syntax:

curl [options] [URL...]

```
[ec2-user@samplegoogle ~]$ curl --help
Usage: curl [options...] <url>
     --abstract-unix-socket <path> Connect via abstract Unix domain socket
     --anyauth       Pick any authentication method
 -a, --append        Append to target file when uploading
     --basic         Use HTTP Basic Authentication
     --cacert <file> CA certificate to verify peer against
     --capath <dir>  CA directory to verify peer against
 -E, --cert <certificate[:password]> Client certificate file and password
     --cert-status   Verify the status of the server certificate
     --cert-type <type> Certificate file type (DER/PEM/ENG)
     --ciphers <list of ciphers> SSL ciphers to use
     --compressed    Request compressed response
     --compressed-ssh Enable SSH compression
 -K, --config <file> Read config from a file
     --connect-timeout <seconds> Maximum time allowed for connection
     --connect-to <HOST1:PORT1:HOST2:PORT2> Connect to host
 -C, --continue-at <offset> Resumed transfer offset
 -b, --cookie <data> Send cookies from string/file
 -c, --cookie-jar <filename> Write cookies to <filename> after operation
     --create-dirs   Create necessary local directory hierarchy
     --crlf          Convert LF to CRLF in upload
     --crlfile <file> Get a CRL list in PEM format from the given file
 -d, --data <data>   HTTP POST data
     --data-ascii <data> HTTP POST ASCII data
     --data-binary <data> HTTP POST binary data
     --data-raw <data> HTTP POST data, '@' allowed
     --data-urlencode <data> HTTP POST data url encoded
```

#curl https://epaper.sakshi.com/

```
[ec2-user@samplegoogle ~]$ curl https://epaper.sakshi.com/
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "https://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="https://www.w3.org/1999/xhtml" >
   <head id="ctl00_Head1">
      <meta http-equiv="X-UA-Compatible" content="IE=edge">
      <meta name="viewport" content="width=device-width, initial-scale=1">
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
      <meta name="description" content="Epaper Sakshi">
      <link rel="shortcut icon" href="https://cache.epapr.in/favicon/logo_image_560a91737de97.jpg" />
      <title> Sakshi Epaper </title>
      <link rel="amphtml" href="https://mpaper.sakshi.com/"/>
      <!-- Latest compiled and minified CSS -->
      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">
      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.4.0/css/font-awesome.min.css">
      <!--        -->
      <link rel=stylesheet href="https://sf.readwhere.com/read/theme/sakshi/css/new_res.min.css?v=1598204012">      <link
rel=stylesheet href="https://sf.readwhere.com/read/styles/wl_login.min.css?v=1580219276">      <link href='https://fonts.g
oogleapis.com/css?family=Open+Sans+Condensed:300,700' rel='stylesheet' type='text/css'>
      <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
      <!-- Smartmetrics Tracking Code -->
      <script type='text/javascript'>
      window.__lo_site_id = 115818;

        (function() {
          var wa = document.createElement('script'); wa.type = 'text/javascript'; wa.async = true;
          wa.src = 'https://d10lpsik1i8c69.cloudfront.net/w.js';
```

wget

wget is the non-interactive network downloader which is used to download files from the server even when the user has not logged on to the system and it can work in the background without hindering the current process.

Syntax :

wget [option] [URL]

wget https://downloads.apache.org/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.tar.gz

```
[ec2-user@samplegoogle ~]$ wget https://downloads.apache.org/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.tar.gz
--2020-10-21 09:12:29--  https://downloads.apache.org/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.tar.gz
Resolving downloads.apache.org (downloads.apache.org)... 88.99.95.219, 2a01:4f8:10a:201a::2
Connecting to downloads.apache.org (downloads.apache.org)|88.99.95.219|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9506321 (9.1M) [application/x-gzip]
Saving to: 'apache-maven-3.6.3-bin.tar.gz'

apache-maven-3.6.3-bin.tar.gz  100%[=================================================>]   9.07M  6.51MB/s    in 1.4s

2020-10-21 09:12:31 (6.51 MB/s) - 'apache-maven-3.6.3-bin.tar.gz' saved [9506321/9506321]

[ec2-user@samplegoogle ~]$ ls -lrt apache*
-rw-rw-r--. 1 ec2-user ec2-user 9506321 Nov 19  2019 apache-maven-3.6.3-bin.tar.gz
[ec2-user@samplegoogle ~]$
```

How to find Linux Flavors' and Version

#cat /etc/os-release

```
[ec2-user@samplegoogle ~]$ cat /etc/os-release
NAME="Red Hat Enterprise Linux"
VERSION="8.2 (Ootpa)"
ID="rhel"
ID_LIKE="fedora"
VERSION_ID="8.2"
PLATFORM_ID="platform:el8"
PRETTY_NAME="Red Hat Enterprise Linux 8.2 (Ootpa)"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:redhat:enterprise_linux:8.2:GA"
HOME_URL="https://www.redhat.com/"
BUG_REPORT_URL="https://bugzilla.redhat.com/"

REDHAT_BUGZILLA_PRODUCT="Red Hat Enterprise Linux 8"
REDHAT_BUGZILLA_PRODUCT_VERSION=8.2
REDHAT_SUPPORT_PRODUCT="Red Hat Enterprise Linux"
REDHAT_SUPPORT_PRODUCT_VERSION="8.2"
[ec2-user@samplegoogle ~]$
```

#cat /etc/*release

```
[ec2-user@samplegoogle ~]$ cat /etc/*release
NAME="Red Hat Enterprise Linux"
VERSION="8.2 (Ootpa)"
ID="rhel"
ID_LIKE="fedora"
VERSION_ID="8.2"
PLATFORM_ID="platform:el8"
PRETTY_NAME="Red Hat Enterprise Linux 8.2 (Ootpa)"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:redhat:enterprise_linux:8.2:GA"
HOME_URL="https://www.redhat.com/"
BUG_REPORT_URL="https://bugzilla.redhat.com/"

REDHAT_BUGZILLA_PRODUCT="Red Hat Enterprise Linux 8"
REDHAT_BUGZILLA_PRODUCT_VERSION=8.2
REDHAT_SUPPORT_PRODUCT="Red Hat Enterprise Linux"
REDHAT_SUPPORT_PRODUCT_VERSION="8.2"
Red Hat Enterprise Linux release 8.2 (Ootpa)
Red Hat Enterprise Linux release 8.2 (Ootpa)
```

uname : Print system information.

#uname --help

```
[ec2-user@samplegoogle ~]$ uname --help
Usage: uname [OPTION]...
Print certain system information.  With no OPTION, same as -s.

  -a, --all                print all information, in the following order,
                             except omit -p and -i if unknown:
  -s, --kernel-name        print the kernel name
  -n, --nodename           print the network node hostname
  -r, --kernel-release     print the kernel release
  -v, --kernel-version     print the kernel version
  -m, --machine            print the machine hardware name
  -p, --processor          print the processor type (non-portable)
  -i, --hardware-platform  print the hardware platform (non-portable)
  -o, --operating-system   print the operating system
      --help     display this help and exit
      --version  output version information and exit

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation at: <https://www.gnu.org/software/coreutils/uname>
or available locally via: info '(coreutils) uname invocation'
[ec2-user@samplegoogle ~]$
```

#uname -a
#uname -s
#uname -n
#uname -r
#uname -v
#uname -m
#uname -p
#uname -i
#uname -o

```
[ec2-user@samplegoogle ~]$ uname -a
Linux samplegoogle.com 4.18.0-193.el8.x86_64 #1 SMP Fri Mar 27 14:35:58 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
[ec2-user@samplegoogle ~]$ uname -s
Linux
[ec2-user@samplegoogle ~]$ uname -n
samplegoogle.com
[ec2-user@samplegoogle ~]$ uname -r
4.18.0-193.el8.x86_64
[ec2-user@samplegoogle ~]$ uname -v
#1 SMP Fri Mar 27 14:35:58 UTC 2020
[ec2-user@samplegoogle ~]$ uname -m
x86_64
[ec2-user@samplegoogle ~]$ uname -p
x86_64
[ec2-user@samplegoogle ~]$ uname -i
x86_64
[ec2-user@samplegoogle ~]$ uname -o
GNU/Linux
[ec2-user@samplegoogle ~]$
```

Printing kernel version

#cat /proc/version

```
[ec2-user@samplegoogle ~]$ cat /proc/version
Linux version 4.18.0-193.el8.x86_64 (mockbuild@x86-vm-08.build.eng.bos.redhat.com) (gcc version 8.3.1 20191121 (Red Hat 8.
3.1-5) (GCC)) #1 SMP Fri Mar 27 14:35:58 UTC 2020
[ec2-user@samplegoogle ~]$
```

To find physical RAM

#cat /proc/meminfo

```
[ec2-user@samplegoogle ~]$ cat /proc/meminfo
MemTotal:         835572 kB
MemFree:          271980 kB
MemAvailable:     543972 kB
Buffers:            2104 kB
Cached:           374424 kB
SwapCached:            0 kB
Active:           248836 kB
Inactive:         183640 kB
Active(anon):      56548 kB
Inactive(anon):    11764 kB
Active(file):     192288 kB
Inactive(file):   171876 kB
Unevictable:           0 kB
Mlocked:               0 kB
SwapTotal:             0 kB
SwapFree:              0 kB
Dirty:                 0 kB
Writeback:             0 kB
AnonPages:         55988 kB
Mapped:            76848 kB
```

#dmesg | grep mem

```
[ec2-user@samplegoogle ~]$ dmesg | grep mem
[    0.000000] BIOS-e820: [mem 0x0000000000000000-0x000000000009dfff] usable
[    0.000000] BIOS-e820: [mem 0x000000000009e000-0x000000000009ffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000000e0000-0x00000000000fffff] reserved
[    0.000000] BIOS-e820: [mem 0x0000000000100000-0x000000003fffffff] usable
[    0.000000] BIOS-e820: [mem 0x00000000fc000000-0x00000000ffffffff] reserved
[    0.000000] e820: update [mem 0x00000000-0x00000fff] usable ==> reserved
[    0.000000] e820: remove [mem 0x000a0000-0x000fffff] usable
[    0.000000] found SMP MP-table at [mem 0x000fbc50-0x000fbc5f]
[    0.000000] kexec: Reserving the low 1M of memory for crashkernel
[    0.000000] RAMDISK: [mem 0x33409000-0x359fcfff]
[    0.000000] Faking a node at [mem 0x0000000000000000-0x000000003fffffff]
[    0.000000] NODE_DATA(0) allocated [mem 0x3ffd6000-0x3fffffff]
[    0.000000] Reserving 160MB of memory at 656MB for crashkernel (System RAM: 1023MB)
[    0.000000]   DMA      [mem 0x0000000000001000-0x0000000000ffffff]
[    0.000000]   DMA32    [mem 0x0000000001000000-0x000000003fffffff]
[    0.000000] Early memory node ranges
[    0.000000]   node   0: [mem 0x0000000000001000-0x000000000009dfff]
[    0.000000]   node   0: [mem 0x0000000000100000-0x000000003fffffff]
[    0.000000] Initmem setup node 0 [mem 0x0000000000001000-0x000000003fffffff]
[    0.000000]   DMA zone: 64 pages used for memmap
[    0.000000]   DMA32 zone: 4032 pages used for memmap
[    0.000000] PM: Registered nosave memory: [mem 0x00000000-0x00000fff]
[    0.000000] PM: Registered nosave memory: [mem 0x0009e000-0x0009ffff]
[    0.000000] PM: Registered nosave memory: [mem 0x000a0000-0x000dffff]
[    0.000000] PM: Registered nosave memory: [mem 0x000e0000-0x000fffff]
[    0.000000] [mem 0x40000000-0xfbffffff] available for PCI devices
```

last: Show listing of last logged in users

```
[ec2-user@samplegoogle ~]$ last --help

Usage:
 last [options] [<username>...] [<tty>...]

Show a listing of last logged in users.

Options:
 -<number>            how many lines to show
 -a, --hostlast       display hostnames in the last column
 -d, --dns            translate the IP number back into a hostname
 -f, --file <file>    use a specific file instead of /var/log/wtmp
 -F, --fulltimes      print full login and logout times and dates
 -i, --ip             display IP numbers in numbers-and-dots notation
 -n, --limit <number> how many lines to show
 -R, --nohostname     don't display the hostname field
 -s, --since <time>   display the lines since the specified time
 -t, --until <time>   display the lines until the specified time
 -p, --present <time> display who were present at the specified time
 -w, --fullnames      display full user and domain names
 -x, --system         display system shutdown entries and run level changes
     --time-format <format>  show timestamps in the specified <format>:
                               notime|short|full|iso

 -h, --help           display this help
 -V, --version        display version

For more details see last(1).
[ec2-user@samplegoogle ~]$
```

#last –w

```
[ec2-user@samplegoogle ~]$ last -w
ec2-user pts/1        157.49.104.231   Wed Oct 21 09:22 - 09:28  (00:05)
ec2-user pts/0        157.49.104.231   Wed Oct 21 09:19   still logged in
ec2-user pts/0        157.49.104.231   Wed Oct 21 09:00 - 09:17  (00:17)
ec2-user pts/0        157.49.104.231   Wed Oct 21 08:35 - 08:49  (00:14)
ec2-user pts/0        157.49.104.231   Wed Oct 21 07:57 - 08:02  (00:05)
reboot   system boot  4.18.0-193.el8.x86_64 Wed Oct 21 07:56   still running
ec2-user pts/0        157.45.11.35     Tue Oct 20 13:08 - 14:17  (01:09)
ec2-user pts/0        157.45.11.35     Tue Oct 20 12:51 - 13:07  (00:16)
ec2-user pts/0        157.45.11.35     Tue Oct 20 12:50 - 12:51  (00:01)
ec2-user pts/4        157.45.25.165    Tue Oct 20 05:45 - 07:59  (02:13)
ec2-user pts/3        157.45.25.165    Tue Oct 20 05:34 - 08:01  (02:27)
ec2-user pts/2        157.45.25.165    Tue Oct 20 05:33 - 07:48  (02:15)
ec2-user pts/1        157.45.25.165    Tue Oct 20 05:25 - 07:40  (02:15)
ec2-user pts/0        157.45.25.165    Tue Oct 20 05:00 - 07:27  (02:27)
reboot   system boot  4.18.0-193.el8.x86_64 Tue Oct 20 05:00 - 14:18  (09:17)
ec2-user pts/0        157.45.6.38      Mon Oct 19 16:45 - 16:54  (00:08)
ec2-user pts/6        157.45.6.38      Mon Oct 19 15:40 - 16:54  (01:14)
ec2-user pts/5        157.45.6.38      Mon Oct 19 15:03 - 16:54  (01:50)
ec2-user pts/4        157.45.6.38      Mon Oct 19 14:49 - 16:54  (02:04)
ec2-user pts/3        157.45.6.38      Mon Oct 19 14:13 - 16:37  (02:24)
ec2-user pts/2        157.45.6.38      Mon Oct 19 13:52 - 16:18  (02:25)
ec2-user pts/1        157.45.6.38      Mon Oct 19 13:47 - 16:13  (02:25)
ec2-user pts/0        157.45.6.38      Mon Oct 19 13:43 - 15:58  (02:14)
ec2-user pts/5        157.45.27.138    Mon Oct 19 11:32 - 11:41  (00:08)
ec2-user pts/4        157.45.27.138    Mon Oct 19 11:25 - 13:38  (02:13)
ec2-user pts/3        157.45.27.138    Mon Oct 19 11:12 - 13:27  (02:14)
ec2-user pts/2        157.45.27.138    Mon Oct 19 10:54 - 13:16  (02:21)
ec2-user pts/1        157.45.27.138    Mon Oct 19 10:04 - 12:35  (02:30)
```

Finding the shells

#cat /etc/shells

#echo $SHELL

```
[ec2-user@samplegoogle ~]$ cat /etc/shells
/bin/sh
/bin/bash
/usr/bin/sh
/usr/bin/bash
[ec2-user@samplegoogle ~]$ echo $SHELL
/bin/bash
[ec2-user@samplegoogle ~]$
```

netstat

Netstat command displays various network related information such as network connections, routing tables, interface statistics, masquerade connections, multicast memberships etc.

```
[ec2-user@samplegoogle ~]$ netstat --help
usage: netstat [-vWeenNcCF] [<Af>] -r         netstat {-V|--version|-h|--help}
       netstat [-vWnNcaeol] [<Socket> ...]
       netstat { [-vWeenNac] -I[<Iface>] | [-veenNac] -i | [-cnNe] -M | -s [-6tuw] } [delay]

         -r, --route              display routing table
         -I, --interfaces=<Iface> display interface table for <Iface>
         -i, --interfaces         display interface table
         -g, --groups             display multicast group memberships
         -s, --statistics         display networking statistics (like SNMP)
         -M, --masquerade         display masqueraded connections

         -v, --verbose            be verbose
         -W, --wide               don't truncate IP addresses
         -n, --numeric            don't resolve names
         --numeric-hosts          don't resolve host names
         --numeric-ports          don't resolve port names
         --numeric-users          don't resolve user names
         -N, --symbolic           resolve hardware names
         -e, --extend             display other/more information
         -p, --programs           display PID/Program name for sockets
         -o, --timers             display timers
         -c, --continuous         continuous listing

         -l, --listening          display listening server sockets
         -a, --all                display all sockets (default: connected)
         -F, --fib                display Forwarding Information Base (default)
         -C, --cache              display routing cache instead of FIB
         -Z, --context            display SELinux security context for sockets

  <Socket>={-t|--tcp} {-u|--udp} {-U|--udplite} {-S|--sctp} {-w|--raw}
           {-x|--unix} --ax25 --ipx --netrom
  <AF>=Use '-6|-4' or '-A <af>' or '--<af>'; default: inet
  List of possible address families (which support routing):
    inet (DARPA Internet) inet6 (IPv6) ax25 (AMPR AX.25)
    netrom (AMPR NET/ROM) ipx (Novell IPX) ddp (Appletalk DDP)
    x25 (CCITT X.25)
[ec2-user@samplegoogle ~]$
```

#netstat –tupl

```
[ec2-user@samplegoogle ~]$ netstat -tupl
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
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:ssh             0.0.0.0:*               LISTEN      -
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN      -
udp        0      0 localhost:323           0.0.0.0:*                           -
udp6       0      0 localhost:323           [::]:*                              -
[ec2-user@samplegoogle ~]$
```

-a -all : Show both listening and non-listening sockets.

#netstat -all

```
[ec2-user@samplegoogle ~]$ netstat -all
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
tcp        0     36 ip-172-31-31-192.us:ssh 157.49.104.231:20737    ESTABLISHED
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
udp        0      0 localhost:323           0.0.0.0:*
udp        0      0 ip-172-31-31-192:bootpc ip-172-31-16-1.u:bootps ESTABLISHED
udp6       0      0 localhost:323           [::]:*
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ACC ]     STREAM     LISTENING     20255    /var/run/.heim_org.h5l.kcm-socket
unix  2      [ ACC ]     STREAM     LISTENING     18472    /run/systemd/private
unix  3      [ ]         DGRAM                    11819    /run/systemd/notify
unix  2      [ ]         DGRAM                    11821    /run/systemd/cgroups-agent
unix  2      [ ACC ]     STREAM     LISTENING     21659    /var/lib/sss/pipes/nss
unix  10     [ ]         DGRAM                    11837    /run/systemd/journal/dev-log
unix  2      [ ACC ]     STREAM     LISTENING     11842    /run/systemd/journal/stdout
unix  6      [ ]         DGRAM                    11845    /run/systemd/journal/socket
unix  2      [ ACC ]     SEQPACKET  LISTENING     18521    /run/udev/control
unix  2      [ ACC ]     STREAM     LISTENING     19807    /run/dbus/system_bus_socket
unix  2      [ ACC ]     STREAM     LISTENING     21568    /var/lib/sss/pipes/private/sbus-monitor
unix  2      [ ACC ]     STREAM     LISTENING     21606    /var/lib/sss/pipes/private/sbus-dp_implicit_files.724
unix  2      [ ACC ]     SEQPACKET  LISTENING     18843    /run/systemd/coredump
unix  2      [ ]         DGRAM                    21158    /var/run/chrony/chronyd.sock
unix  2      [ ]         DGRAM                    38612    /run/user/1000/systemd/notify
unix  2      [ ACC ]     STREAM     LISTENING     38616    /run/user/1000/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     38624    /run/user/1000/bus
unix  2      [ ]         STREAM     CONNECTED     23156
unix  3      [ ]         STREAM     CONNECTED     20941    /run/systemd/journal/stdout
```

List all tcp ports

netstat –at

#netstat -lt

```
[ec2-user@samplegoogle ~]$ netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
tcp        0     36 ip-172-31-31-192.us:ssh 157.49.104.231:20737    ESTABLISHED
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
[ec2-user@samplegoogle ~]$
```

```
[ec2-user@samplegoogle ~]$ netstat -lt
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
[ec2-user@samplegoogle ~]$
```

List all udp ports

#netstat –au

#netstat -lu

```
[ec2-user@samplegoogle ~]$ netstat -au
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 localhost:323           0.0.0.0:*
udp        0      0 ip-172-31-31-192:bootpc ip-172-31-16-1.u:bootps ESTABLISHED
udp6       0      0 localhost:323           [::]:*
[ec2-user@samplegoogle ~]$
```

```
[ec2-user@samplegoogle ~]$ netstat -lu
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 localhost:323           0.0.0.0:*
udp6       0      0 localhost:323           [::]:*
[ec2-user@samplegoogle ~]$
```

List only listening ports

#netstat –l

```
[ec2-user@samplegoogle ~]$ netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
udp        0      0 localhost:323           0.0.0.0:*
udp6       0      0 localhost:323           [::]:*
Active UNIX domain sockets (only servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ACC ]     STREAM     LISTENING     20255    /var/run/.heim_org.h5l.kcm-socket
unix  2      [ ACC ]     STREAM     LISTENING     18472    /run/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     21659    /var/lib/sss/pipes/nss
unix  2      [ ACC ]     STREAM     LISTENING     11842    /run/systemd/journal/stdout
unix  2      [ ACC ]     SEQPACKET  LISTENING     18521    /run/udev/control
unix  2      [ ACC ]     STREAM     LISTENING     19807    /run/dbus/system_bus_socket
unix  2      [ ACC ]     STREAM     LISTENING     21568    /var/lib/sss/pipes/private/sbus-monitor
unix  2      [ ACC ]     STREAM     LISTENING     21606    /var/lib/sss/pipes/private/sbus-dp_implicit_files.724
unix  2      [ ACC ]     SEQPACKET  LISTENING     18843    /run/systemd/coredump
unix  2      [ ACC ]     STREAM     LISTENING     38616    /run/user/1000/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     38624    /run/user/1000/bus
Active Bluetooth connections (only servers)
Proto  Destination       Source            State         PSM DCID   SCID      IMTU    OMTU Security
Proto  Destination       Source            State     Channel
[ec2-user@samplegoogle ~]$
```

List only the listening UNIX ports

#netstat -lx

```
[ec2-user@samplegoogle ~]$ netstat -lx
Active UNIX domain sockets (only servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ACC ]     STREAM     LISTENING     20255    /var/run/.heim_org.h5l.kcm-socket
unix  2      [ ACC ]     STREAM     LISTENING     18472    /run/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     21659    /var/lib/sss/pipes/nss
unix  2      [ ACC ]     STREAM     LISTENING     11842    /run/systemd/journal/stdout
unix  2      [ ACC ]     SEQPACKET  LISTENING     18521    /run/udev/control
unix  2      [ ACC ]     STREAM     LISTENING     19807    /run/dbus/system_bus_socket
unix  2      [ ACC ]     STREAM     LISTENING     21568    /var/lib/sss/pipes/private/sbus-monitor
unix  2      [ ACC ]     STREAM     LISTENING     21606    /var/lib/sss/pipes/private/sbus-dp_implicit_files.724
unix  2      [ ACC ]     SEQPACKET  LISTENING     18843    /run/systemd/coredump
unix  2      [ ACC ]     STREAM     LISTENING     38616    /run/user/1000/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     38624    /run/user/1000/bus
[ec2-user@samplegoogle ~]$
```

List the statistics for all ports

#netstat -s

```
[ec2-user@samplegoogle ~]$ netstat -s
Ip:
    Forwarding: 2
    9501 total packets received
    4 with invalid addresses
    0 forwarded
    0 incoming packets discarded
    9497 incoming packets delivered
    8214 requests sent out
    246 dropped because of missing route
Icmp:
    6 ICMP messages received
    2 input ICMP message failed
    ICMP input histogram:
        destination unreachable: 2
        echo requests: 4
```

List the statistics for TCP (or) UDP ports

#netstat -st

```
[ec2-user@samplegoogle ~]$ netstat -st
IcmpMsg:
    InType3: 2
    InType8: 4
    OutType0: 4
    OutType3: 37
Tcp:
    133 active connection openings
    78 passive connection openings
    7 failed connection attempts
    57 connection resets received
    2 connections established
    8410 segments received
    7287 segments sent out
    18 segments retransmitted
    0 bad segments received
    1351 resets sent
UdpLite:
TcpExt:
    3 resets received for embryonic SYN_RECV sockets
    60 TCP sockets finished time wait in fast timer
    145 delayed acks sent
    2761 packet headers predicted
    526 acknowledgments not containing data payload received
    1388 predicted acknowledgments
    1 congestion windows recovered without slow start after partial ack
    TCPTimeouts: 12
    TCPLossProbes: 5
    TCPDSACKRecv: 1
    1 connections reset due to unexpected data
    44 connections reset due to early user close
```

#netstat -su

```
[ec2-user@samplegoogle ~]$ netstat -su
IcmpMsg:
    InType3: 2
    InType8: 4
    OutType0: 4
    OutType3: 37
Udp:
    1100 packets received
    37 packets to unknown port received
    0 packet receive errors
    1100 packets sent
    0 receive buffer errors
    0 send buffer errors
UdpLite:
IpExt:
    InOctets: 23969665
    OutOctets: 1203325
    InNoECTPkts: 22956
    InECT0Pkts: 9
[ec2-user@samplegoogle ~]$
```

Display PID and program names in the output

#netstat -pt

```
[ec2-user@samplegoogle ~]$ netstat -pt
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0     36 ip-172-31-31-192.us:ssh 157.49.104.231:13112    ESTABLISHED -
[ec2-user@samplegoogle ~]$
```

Print the netstat information continuously

#netstat –c

```
[ec2-user@samplegoogle ~]$ netstat -c
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0     36 ip-172-31-31-192.us:ssh 157.49.104.231:13112    ESTABLISHED
udp        0      0 ip-172-31-31-192:bootpc ip-172-31-16-1.u:bootps ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  3      [ ]         DGRAM                    11819    /run/systemd/notify
unix  2      [ ]         DGRAM                    11821    /run/systemd/cgroups-agent
unix  10     [ ]         DGRAM                    11837    /run/systemd/journal/dev-log
unix  6      [ ]         DGRAM                    11845    /run/systemd/journal/socket
unix  2      [ ]         DGRAM                    21158    /var/run/chrony/chronyd.sock
unix  2      [ ]         DGRAM                    42694    /run/user/1000/systemd/notify
unix  2      [ ]         STREAM     CONNECTED     23156
unix  3      [ ]         STREAM     CONNECTED     20941    /run/systemd/journal/stdout
unix  2      [ ]         DGRAM                    24617
unix  3      [ ]         STREAM     CONNECTED     23398    /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     21547
unix  3      [ ]         STREAM     CONNECTED     21631
unix  3      [ ]         STREAM     CONNECTED     21845
unix  2      [ ]         STREAM     CONNECTED     34852
unix  3      [ ]         STREAM     CONNECTED     21657
unix  3      [ ]         STREAM     CONNECTED     24518    /run/systemd/journal/stdout
```

Non-supportive address families in the system

netstat –verbose

```
[ec2-user@samplegoogle ~]$ netstat --verbose
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0     36 ip-172-31-31-192.us:ssh 106.197.193.32:59298    ESTABLISHED
tcp        0      0 ip-172-31-31-192.us:ssh 106.197.193.32:28149    ESTABLISHED
netstat: no support for `AF INET (sctp)' on this system.
netstat: no support for `AF INET (sctp)' on this system.
udp        0      0 ip-172-31-31-192:bootpc ip-172-31-16-1.u:bootps ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ]         DGRAM                    44052    /run/user/1000/systemd/notify
unix  3      [ ]         DGRAM                    11819    /run/systemd/notify
unix  2      [ ]         DGRAM                    11821    /run/systemd/cgroups-agent
unix  11     [ ]         DGRAM                    11837    /run/systemd/journal/dev-log
unix  6      [ ]         DGRAM                    11845    /run/systemd/journal/socket
unix  2      [ ]         DGRAM                    21158    /var/run/chrony/chronyd.sock
unix  2      [ ]         STREAM     CONNECTED     23156
unix  3      [ ]         STREAM     CONNECTED     20941    /run/systemd/journal/stdout
unix  3      [ ]         DGRAM                    44054
unix  2      [ ]         DGRAM                    24617
unix  3      [ ]         STREAM     CONNECTED     44243    /var/lib/sss/pipes/nss
unix  3      [ ]         STREAM     CONNECTED     44028
unix  3      [ ]         STREAM     CONNECTED     23398    /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     21547
unix  3      [ ]         STREAM     CONNECTED     21631
unix  3      [ ]         STREAM     CONNECTED     21845
unix  2      [ ]         STREAM     CONNECTED     34852
unix  3      [ ]         STREAM     CONNECTED     21657
unix  3      [ ]         STREAM     CONNECTED     24518    /run/systemd/journal/stdout
unix  3      [ ]         DGRAM                    19718
```

Kernel routing information.

#netstat -r

```
[ec2-user@samplegoogle ~]$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         ip-172-31-16-1. 0.0.0.0         UG        0 0          0 eth0
ip-172-31-16-0. 0.0.0.0         255.255.240.0   U         0 0          0 eth0
[ec2-user@samplegoogle ~]$
```

The port on which a program is running

#netstat -ap | grep ssh

```
[ec2-user@samplegoogle ~]$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         ip-172-31-16-1. 0.0.0.0         UG        0 0          0 eth0
ip-172-31-16-0. 0.0.0.0         255.255.240.0   U         0 0          0 eth0
[ec2-user@samplegoogle ~]$ netstat -ap | grep ssh
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN      -
tcp        0     36 ip-172-31-31-192.us:ssh 157.49.104.231:18963    ESTABLISHED -
tcp        0      0 ip-172-31-31-192.us:ssh 106.197.193.32:59298    ESTABLISHED -
tcp        0      0 ip-172-31-31-192.us:ssh 106.197.193.32:28149    ESTABLISHED -
tcp        0      0 ip-172-31-31-192.us:ssh 157.49.104.231:19623    ESTABLISHED -
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN      -
[ec2-user@samplegoogle ~]$
```

Which process is using a particular port

#netstat -an | grep ':22'

```
[ec2-user@samplegoogle ~]$ netstat -an | grep ':22'
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0     36 172.31.31.192:22        157.49.104.231:18963    ESTABLISHED
tcp        0      0 172.31.31.192:22        106.197.193.32:59298    ESTABLISHED
tcp        0      0 172.31.31.192:22        106.197.193.32:28149    ESTABLISHED
tcp        0      0 172.31.31.192:22        157.49.104.231:19623    ESTABLISHED
tcp6       0      0 :::22                   :::*                    LISTEN
[ec2-user@samplegoogle ~]$
```

List of network interfaces

#netstat –i

#netstat -ie

```
[ec2-user@samplegoogle ~]$ netstat -i
Kernel Interface table
Iface             MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0             9001    24266      0      0 0          9481      0      0      0 BMRU
lo              65536        8      0      0 0             8      0      0      0 LRU
[ec2-user@samplegoogle ~]$
```

```
[ec2-user@samplegoogle ~]$ netstat -ie
Kernel Interface table
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
        inet 172.31.31.192  netmask 255.255.240.0  broadcast 172.31.31.255
        inet6 fe80::48f:ccff:fe51:8f92  prefixlen 64  scopeid 0x20<link>
        ether 06:8f:cc:51:8f:92  txqueuelen 1000  (Ethernet)
        RX packets 24297  bytes 24752283 (23.6 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9507  bytes 1427663 (1.3 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 400 (400.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 400 (400.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[ec2-user@samplegoogle ~]$
```

cal

If a user wants a quick view of calendar in Linux terminal, cal is the command for you. By default, cal command shows current month calendar as output.
cal command is a calendar command in Linux which is used to see the calendar of a specific month or a whole year.

Syntax:
cal [[month] year]

cal: Shows current month calendar on the terminal.
#cal

```
[ec2-user@samplegoogle ~]$ cal
    October 2020
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

[ec2-user@samplegoogle ~]$
```

cal 111986 : Shows calendar of selected month and year.
#cal 11 1986

```
[ec2-user@samplegoogle ~]$ cal 11 1986
    November 1986
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
[ec2-user@samplegoogle ~]$
```

cal 2020 : Shows the whole calendar of the year

```
[ec2-user@samplegoogle ~]$ cal 2020
                                2020

       January               February                March
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
          1  2  3  4                      1      1  2  3  4  5  6  7
 5  6  7  8  9 10 11    2  3  4  5  6  7  8      8  9 10 11 12 13 14
12 13 14 15 16 17 18    9 10 11 12 13 14 15     15 16 17 18 19 20 21
19 20 21 22 23 24 25   16 17 18 19 20 21 22     22 23 24 25 26 27 28
26 27 28 29 30 31      23 24 25 26 27 28 29     29 30 31
                       30

       April                  May                   June
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
          1  2  3  4                   1  2         1  2  3  4  5  6
 5  6  7  8  9 10 11    3  4  5  6  7  8  9      7  8  9 10 11 12 13
12 13 14 15 16 17 18   10 11 12 13 14 15 16     14 15 16 17 18 19 20
19 20 21 22 23 24 25   17 18 19 20 21 22 23     21 22 23 24 25 26 27
26 27 28 29 30         24 25 26 27 28 29 30     28 29 30
                       31

       July                  August                September
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
          1  2  3  4                      1         1  2  3  4  5
 5  6  7  8  9 10 11    2  3  4  5  6  7  8      6  7  8  9 10 11 12
12 13 14 15 16 17 18    9 10 11 12 13 14 15     13 14 15 16 17 18 19
19 20 21 22 23 24 25   16 17 18 19 20 21 22     20 21 22 23 24 25 26
26 27 28 29 30 31      23 24 25 26 27 28 29     27 28 29 30
                       30 31

       October               November              December
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
             1  2  3      1  2  3  4  5  6  7         1  2  3  4  5
 4  5  6  7  8  9 10      8  9 10 11 12 13 14      6  7  8  9 10 11 12
11 12 13 14 15 16 17     15 16 17 18 19 20 21     13 14 15 16 17 18 19
18 19 20 21 22 23 24     22 23 24 25 26 27 28     20 21 22 23 24 25 26
25 26 27 28 29 30 31     29 30                    27 28 29 30 31

[ec2-user@samplegoogle ~]$
```

cal 2018 | more : But year may not be visible in the same screen use more with cal use spacebar to scroll down.

#cal 2018 | more

```
[ec2-user@samplegoogle ~]$ cal 2018 | more
                                2018

       January               February                March
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6                   1  2  3               1  2  3
 7  8  9 10 11 12 13    4  5  6  7  8  9 10      4  5  6  7  8  9 10
14 15 16 17 18 19 20   11 12 13 14 15 16 17     11 12 13 14 15 16 17
21 22 23 24 25 26 27   18 19 20 21 22 23 24     18 19 20 21 22 23 24
28 29 30 31            25 26 27 28              25 26 27 28 29 30 31

       April                  May                   June
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7            1  2  3  4  5                  1  2
 8  9 10 11 12 13 14    6  7  8  9 10 11 12      3  4  5  6  7  8  9
15 16 17 18 19 20 21   13 14 15 16 17 18 19     10 11 12 13 14 15 16
22 23 24 25 26 27 28   20 21 22 23 24 25 26     17 18 19 20 21 22 23
29 30                  27 28 29 30 31           24 25 26 27 28 29 30

       July                  August                September
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7               1  2  3  4                     1
 8  9 10 11 12 13 14    5  6  7  8  9 10 11      2  3  4  5  6  7  8
15 16 17 18 19 20 21   12 13 14 15 16 17 18      9 10 11 12 13 14 15
22 23 24 25 26 27 28   19 20 21 22 23 24 25     16 17 18 19 20 21 22
29 30 31               26 27 28 29 30 31        23 24 25 26 27 28 29
                                                30

       October               November              December
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6                   1  2  3                     1
 7  8  9 10 11 12 13    4  5  6  7  8  9 10      2  3  4  5  6  7  8
14 15 16 17 18 19 20   11 12 13 14 15 16 17      9 10 11 12 13 14 15
21 22 23 24 25 26 27   18 19 20 21 22 23 24     16 17 18 19 20 21 22
28 29 30 31            25 26 27 28 29 30        23 24 25 26 27 28 29
                                                30 31
[ec2-user@samplegoogle ~]$
```

cal -3 : Shows calendar of previous, current and next month.

```
[ec2-user@samplegoogle ~]$ cal -3
   September 2020         October 2020          November 2020
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
       1  2  3  4  5                 1  2  3   1  2  3  4  5  6  7
 6  7  8  9 10 11 12   4  5  6  7  8  9 10   8  9 10 11 12 13 14
13 14 15 16 17 18 19  11 12 13 14 15 16 17  15 16 17 18 19 20 21
20 21 22 23 24 25 26  18 19 20 21 22 23 24  22 23 24 25 26 27 28
27 28 29 30           25 26 27 28 29 30 31  29 30

[ec2-user@samplegoogle ~]$
```

History

history command is used to view the previously executed command. This feature was not available in the Bourne shell. Bash and Korn support this feature in which every command executed is treated as the event and is associated with an event number using which they can be recalled and changed if required. These commands are saved in a history file. In Bash shell history command shows the whole list of the command.

#history

```
[ec2-user@samplegoogle ~]$ history
   11  ll
   12  chmod -R 777 sample
   13  ll
   14  cd sample
   15  clear
   16  touch tst.txt
   17  ll
   18  chmod u-rwx tst.txt
   19  ll
   20  chmod u+rwx tst.txt
   21  ll
   22  chmod g-rwx tst.txt
   23  ll
   24  chmod go-rwx tst.txt
   25  ll
   26  chmod go+rx tst.txt
   27  ll
   28  clear
```

To show the limited number of commands

#history 5

```
[ec2-user@samplegoogle ~]$ history 5
 1007  cal -3
 1008  clock
 1009  clock --verbose
 1010  history
 1011  history 5
[ec2-user@samplegoogle ~]$
```

Uptime

It is used to find out how long the system is active (running). This command returns set of values that involve, the current time, and the amount of time system is in running state, number of users currently logged into, and the load time for the past 1, 5 and 15 minutes respectively.

Syntax:

uptime [-options]

#uptime

```
[ec2-user@samplegoogle ~]$ uptime
 12:13:07 up  4:16,  3 users,  load average: 0.00, 0.00, 0.00
[ec2-user@samplegoogle ~]$
```

uptime in human-readable format

#uptime -p

```
[ec2-user@samplegoogle ~]$ uptime -p
up 4 hours, 18 minutes
[ec2-user@samplegoogle ~]$
```

starting time/specified time when the system started

#uptime -s

```
[ec2-user@samplegoogle ~]$ uptime -s
2020-10-21 07:56:34
[ec2-user@samplegoogle ~]$
```

ping command

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.

PING Version:

To get ping version installed on your system.

#sudo ping -v

```
[ec2-user@samplegoogle ~]$ ping -V
ping utility, iputils-s20180629
[ec2-user@samplegoogle ~]$
```

Using PING:

#ping google.com

```
[ec2-user@samplegoogle ~]$ ping google.com
PING google.com (172.217.9.46) 56(84) bytes of data.
64 bytes from ord38s08-in-f14.1e100.net (172.217.9.46): icmp_seq=1 ttl=100 time=16.8 ms
64 bytes from ord38s08-in-f14.1e100.net (172.217.9.46): icmp_seq=2 ttl=100 time=16.9 ms
64 bytes from ord38s08-in-f14.1e100.net (172.217.9.46): icmp_seq=3 ttl=100 time=16.9 ms
64 bytes from ord38s08-in-f14.1e100.net (172.217.9.46): icmp_seq=4 ttl=100 time=16.9 ms
64 bytes from ord38s08-in-f14.1e100.net (172.217.9.46): icmp_seq=5 ttl=100 time=16.8 ms
```

Controlling the number of pings:

#ping -c 5 google.com

```
[ec2-user@samplegoogle ~]$ ping -c 5 google.com
PING google.com (172.217.9.46) 56(84) bytes of data.
64 bytes from ord38s08-in-f14.1e100.net (172.217.9.46): icmp_seq=1 ttl=100 time=16.8 ms
64 bytes from ord38s08-in-f14.1e100.net (172.217.9.46): icmp_seq=2 ttl=100 time=16.9 ms
64 bytes from ord38s08-in-f14.1e100.net (172.217.9.46): icmp_seq=3 ttl=100 time=16.9 ms
64 bytes from ord38s08-in-f14.1e100.net (172.217.9.46): icmp_seq=4 ttl=100 time=17.4 ms
64 bytes from ord38s08-in-f14.1e100.net (172.217.9.46): icmp_seq=5 ttl=100 time=16.9 ms

--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 11ms
rtt min/avg/max/mdev = 16.800/16.993/17.449/0.246 ms
[ec2-user@samplegoogle ~]$
```

Timout PING

#ping -w 3 google.com

```
[ec2-user@samplegoogle ~]$ ping -w 3 google.com
PING google.com (172.217.9.46) 56(84) bytes of data.
64 bytes from ord38s08-in-f14.1e100.net (172.217.9.46): icmp_seq=1 ttl=100 time=16.9 ms
64 bytes from ord38s08-in-f14.1e100.net (172.217.9.46): icmp_seq=2 ttl=100 time=16.8 ms
64 bytes from ord38s08-in-f14.1e100.net (172.217.9.46): icmp_seq=3 ttl=100 time=16.9 ms

--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 6ms
rtt min/avg/max/mdev = 16.840/16.853/16.868/0.011 ms
[ec2-user@samplegoogle ~]$
```

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. Below image depicts how traceroute command is used to reach the Google(172.217.26.206) host from the local machine and it also prints detail about all the hops that it visits in between.

#traceroute -4 google.com

#traceroute -6 google.com

```
[ec2-user@samplegoogle ~]$ traceroute -4 google.com
traceroute to google.com (172.217.4.46), 30 hops max, 60 byte packets
 1  ec2-52-15-0-105.us-east-2.compute.amazonaws.com (52.15.0.105)  11.053 ms ec2-52-15-0-103.us-east-2.compute.amazonaws.com (52.15.0.103)  3.349 ms  3.302 ms
 2  * 100.65.25.16 (100.65.25.16)  4.303 ms 100.65.24.16 (100.65.24.16)  2.241 ms
 3  * 100.66.12.30 (100.66.12.30)  4.625 ms 100.66.12.206 (100.66.12.206)  5.662 ms
 4  100.66.14.74 (100.66.14.74)  12.819 ms 100.66.14.202 (100.66.14.202)  12.714 ms 100.66.14.142 (100.66.14.142)  12.817 ms
 5  100.66.6.11 (100.66.6.11)  17.241 ms 100.66.6.109 (100.66.6.109)  21.756 ms 100.66.7.133 (100.66.7.133)  11.429 ms
 6  * 100.66.4.139 (100.66.4.139)  16.031 ms 100.66.4.121 (100.66.4.121)  20.252 ms
 7  100.65.9.33 (100.65.9.33)  0.515 ms 100.65.10.225 (100.65.10.225)  0.301 ms 100.65.9.33 (100.65.9.33)  0.724 ms
 8  52.95.3.139 (52.95.3.139)  1.151 ms 15.230.39.195 (15.230.39.195)  1.555 ms 15.230.39.221 (15.230.39.221)  0.651 ms
 9  15.230.39.80 (15.230.39.80)  1.004 ms 15.230.39.214 (15.230.39.214)  1.448 ms 15.230.39.90 (15.230.39.90)  1.529 ms
10  52.95.2.211 (52.95.2.211)  0.825 ms 52.95.1.205 (52.95.1.205)  0.664 ms 52.95.2.237 (52.95.2.237)  0.874 ms
11  100.92.53.114 (100.92.53.114)  10.928 ms 100.92.53.152 (100.92.53.152)  16.225 ms 100.92.53.156 (100.92.53.156)  16.187 ms
12  100.92.48.26 (100.92.48.26)  11.828 ms 100.92.43.10 (100.92.43.10)  11.510 ms 100.92.43.100 (100.92.43.100)  11.170 ms
13  100.92.43.89 (100.92.43.89)  10.752 ms 100.92.48.45 (100.92.48.45)  11.436 ms 100.92.48.23 (100.92.48.23)  11.719 ms
14  100.92.49.60 (100.92.49.60)  11.412 ms 100.92.44.38 (100.92.44.38)  11.536 ms 100.92.49.40 (100.92.49.40)  15.701 ms
15  100.92.44.51 (100.92.44.51)  10.883 ms 100.92.44.31 (100.92.44.31)  11.849 ms 100.92.44.51 (100.92.44.51)  10.835 ms
16  52.93.130.134 (52.93.130.134)  11.088 ms 52.93.133.110 (52.93.133.110)  10.805 ms 52.93.132.72 (52.93.132.72)  11.479 ms
17  100.91.168.0 (100.91.168.0)  10.985 ms 100.91.163.58 (100.91.163.58)  11.606 ms 100.91.163.100 (100.91.163.100)  11.099 ms
```

```
[ec2-user@samplegoogle ~]$ traceroute -6 google.com
traceroute to google.com (2607:f8b0:4009:804::200e), 30 hops max, 80 byte packets
connect: Network is unreachable
```

Sort command

SORT command is used to sort a file, arranging the records in a particular order. By default, the sort command sorts file assuming the contents are ASCII. Using options in sort command, it can also be used to sort numerically.

Sorting a file

#sort names.txt

#sort names.txt>ouput.txt

```
[ec2-user@samplegoogle ppreddy]$ cat names.txt
Ram
Raj
Amar
Madhav
Reddy
Deva
Alluri
Sudhakar
Narendra
[ec2-user@samplegoogle ppreddy]$ sort names.txt
Alluri
Amar
Deva
Madhav
Narendra
Raj
Ram
Reddy
Sudhakar
[ec2-user@samplegoogle ppreddy]$
```

```
[ec2-user@samplegoogle ppreddy]$ sort names.txt > output.txt
[ec2-user@samplegoogle ppreddy]$ ll
total 28
-rw-rw-r--. 1 ec2-user ec2-user  50 Oct 22 07:37 employee.txt
-rw-rw-r--. 1 ec2-user ec2-user 465 Oct 20 13:36 mytarfile.tgz
-rw-rw-r--. 1 ec2-user ec2-user  57 Oct 22 07:31 names.txt
-rw-rw-r--. 1 ec2-user ec2-user  33 Oct 22 07:34 numbers.txt
-rw-rw-r--. 1 ec2-user ec2-user  57 Oct 22 07:42 output.txt
-rw-rw-r--. 1 ec2-user ec2-user  50 Oct 15 17:42 sample.txt
-rw-rw-r--. 1 ec2-user ec2-user 315 Oct 16 01:40 test.txt
[ec2-user@samplegoogle ppreddy]$ cat output.txt
Alluri
Amar
Deva
Madhav
Narendra
Raj
Ram
Reddy
Sudhakar
[ec2-user@samplegoogle ppreddy]$
```

Sorting In Reverse Order

#sort -r names.txt

```
[ec2-user@samplegoogle ppreddy]$ sort -r names.txt
Sudhakar
Reddy
Ram
Raj
Narendra
Madhav
Deva
Amar
Alluri
[ec2-user@samplegoogle ppreddy]$
```

Sorting the numeric data and reverse

#sort -n numbers.txt

sort -nr numbers.txt

```
[ec2-user@samplegoogle ppreddy]$ cat numbers.txt
44
11
01
09
1000
888
555
1000000
[ec2-user@samplegoogle ppreddy]$ sort -n numbers.txt
01
09
11
44
555
888
1000
1000000
[ec2-user@samplegoogle ppreddy]$ sort -nr numbers.txt
1000000
1000
888
555
44
11
09
01
[ec2-user@samplegoogle ppreddy]$
```

sorting a table

#sort -k 2n employee.txt

```
[ec2-user@samplegoogle ppreddy]$ cat employee.txt
Rahul  4000
Sekhar 4444
Shyam  100000
Madhu  9000
[ec2-user@samplegoogle ppreddy]$ sort -k 2n employee.txt
Rahul  4000
Sekhar 4444
Madhu  9000
Shyam  100000
[ec2-user@samplegoogle ppreddy]$
```

Test command

The test command is used to check file types and compare values. Test is used in conditional execution.This command is used to see if an expression is true, and if it is true it return zero(0), otherwise returns nonzero for false.

#test 5 -gt 2 && echo "Yes"
#test 1 -lt 2 && echo "Yes"
#test 5 -eq 5 && echo Yes || echo No
#test 5 -eq 15 && echo Yes || echo No
#test 5 -ne 10 && echo Yes || echo No

```
[ec2-user@samplegoogle ~]$ test 5 -gt 2 && echo "Yes"
Yes
[ec2-user@samplegoogle ~]$ test 1 -lt 2 && echo "Yes"
Yes
[ec2-user@samplegoogle ~]$ test 5 -eq 5 && echo Yes || echo No
Yes
[ec2-user@samplegoogle ~]$ test 5 -eq 15 && echo Yes || echo No
No
[ec2-user@samplegoogle ~]$ test 5 -ne 10 && echo Yes || echo No
Yes
[ec2-user@samplegoogle ~]$
```

expr command

The expr command in Unix evaluates a given expression and displays its corresponding output. It is used for:

- Basic operations like addition, subtraction, multiplication, division, and modulus on integers.
- Evaluating regular expressions, string operations like substring, length of strings etc.

Syntax:
$expr expression

#expr 20 + 2
#expr 20 * 2

```
[ec2-user@samplegoogle ~]$ expr --version
expr (GNU coreutils) 8.30
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by Mike Parker, James Youngman, and Paul Eggert.
[ec2-user@samplegoogle ~]$ expr 20 + 2
22
[ec2-user@samplegoogle ~]$ expr 20 * 2
expr: syntax error: unexpected argument ‘700’
[ec2-user@samplegoogle ~]$ expr 20 \* 2
40
[ec2-user@samplegoogle ~]$
```

How to find weather system type is 32 bit or 64 bit?
Below command will give the weather system type is 32 bit or 64 bit
#getconf LONG_BIT

```
[ec2-user@samplegoogle ~]$ getconf LONG_BIT
64
[ec2-user@samplegoogle ~]$
```

cut command

The cut command in UNIX is a command for cutting out the sections from each line of files and writing the result to standard output. It can be used to cut parts of a line by **byte position, character and field**. Basically the cut command slices a line and extracts the text. It is necessary to specify option with command otherwise it gives error.

Syntax:
cut OPTION... [FILE]...
#cat state.txt

```
[ec2-user@samplegoogle ppreddy]$ vi state.txt
[ec2-user@samplegoogle ppreddy]$ cat state.txt
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
Karnataka
Telengana
Tamilnadu
```

-b(byte): To extract the specific bytes, you need to follow -b option with the list of byte numbers separated by comma. Range of bytes can also be specified using the hyphen(-). It is necessary to specify list of byte numbers otherwise it gives error. **Tabs and backspaces** are treated like as a character of 1 byte.

List without ranges
#cut -b 1,2,3 state.txt

```
[ec2-user@samplegoogle ppreddy]$ cut -b 1,2,3 state.txt
And
Aru
Ass
Bih
Chh
Kar
Tel
Tam
```

List with ranges

#cut -b 1-3,5-7 state.txt

```
[ec2-user@samplegoogle ppreddy]$ cut -b 1-3,5-7 state.txt
Andra
Aruach
Assm
Bihr
Chhtti
Karata
Telnga
Tamlna
```

#cut -b 1- state.txt

1- indicate from 1st byte to end byte of a line

```
[ec2-user@samplegoogle ppreddy]$ cut -b 1- state.txt
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
Karnataka
Telengana
Tamilnadu
[ec2-user@samplegoogle ppreddy]$
```

#cut -b -3 state.txt (-3 indicate from 1st byte to 3rd byte of a line)

```
[ec2-user@samplegoogle ppreddy]$ cut -b -3 state.txt
And
Aru
Ass
Bih
Chh
Kar
Tel
Tam
[ec2-user@samplegoogle ppreddy]$
```

-c (column): To cut by character use the -c option. This selects the characters given to the -c option. This can be a list of numbers separated comma or a range of numbers separated by hyphen(-). Tabs and backspaces are treated as a character. It is necessary to specify list of character numbers otherwise it gives error with this option.

Syntax:

$cut -c [(k)-(n)/(k),(n)/(n)] filename

Here,k denotes the starting position of the character and n denotes the ending position of the character in each line, if k and n are separated by "-" otherwise they are only the position of character in each line from the file taken as an input.

#cut -c 2,5,7 state.txt

(command prints second, fifth and seventh character from each line of the file.)

```
[ec2-user@samplegoogle ppreddy]$ cut -c 2,5,7 state.txt
nr
rah
sm
ir
hti
aaa
ena
ala
```

#cut -c 1-7 state.txt (command prints first seven characters of each line from the file)

```
[ec2-user@samplegoogle ppreddy]$ cut -c 1-7 state.txt
Andhra
Arunach
Assam
Bihar
Chhatti
Karnata
Telenga
Tamilna
[ec2-user@samplegoogle ppreddy]$
```

#cut -c 1- state.txt (command prints starting from first character to end. Here in command only starting position is specified and the ending position is omitted.)

```
[ec2-user@samplegoogle ppreddy]$ cut -c 1- state.txt
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
Karnataka
Telengana
Tamilnadu
[ec2-user@samplegoogle ppreddy]$
```

#cut -c -5 state.txt

(command prints starting position to the fifth character. Here the starting position

is omitted and the ending position is specified)

```
[ec2-user@samplegoogle ppreddy]$ cut -c -5 state.txt
Andhr
Aruna
Assam
Bihar
Chhat
Karna
Telen
Tamil
[ec2-user@samplegoogle ppreddy]$
```

PATH Settings

PATH is an *environmental variable* in Linux and other Unix-like operating systems that tells the shell which directories to search for executable files (i.e., ready-to-run programs) in response to commands issued by a user.

1. Download the java in /opt folder

cd /opt

sudo wget https://corretto.aws/downloads/latest/amazon-corretto-8-x64-linux-jdk.tar.gz

```
[ec2-user@samplegoogle opt]$ cd /opt
[ec2-user@samplegoogle opt]$ sudo wget https://corretto.aws/downloads/latest/amazon-corretto-8-x64-linux-jdk.tar.gz
--2020-10-24 13:59:42--  https://corretto.aws/downloads/latest/amazon-corretto-8-x64-linux-jdk.tar.gz
Resolving corretto.aws (corretto.aws)... 13.33.163.19, 13.33.163.93, 13.33.163.125, ...
Connecting to corretto.aws (corretto.aws)|13.33.163.19|:443... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: /downloads/resources/8.272.10.3/amazon-corretto-8.272.10.3-linux-x64.tar.gz [following]
--2020-10-24 13:59:43--  https://corretto.aws/downloads/resources/8.272.10.3/amazon-corretto-8.272.10.3-linux-x64.tar.gz
Reusing existing connection to corretto.aws:443.
HTTP request sent, awaiting response... 200 OK
Length: 117929735 (112M) [binary/octet-stream]
Saving to: 'amazon-corretto-8-x64-linux-jdk.tar.gz'

amazon-corretto-8-x64-linux-jd 100%[=================================================>] 112.47M  65.8MB/s    in 1.7s

2020-10-24 13:59:45 (65.8 MB/s) - 'amazon-corretto-8-x64-linux-jdk.tar.gz' saved [117929735/117929735]

[ec2-user@samplegoogle opt]$
```

2. Un tar and rename the java binaries

sudo tar -xzf amazon-corretto-8-x64-linux-jdk.tar.gz

sudo mv amazon-corretto-8.272.10.3-linux-x64 java-1.8

```
[ec2-user@samplegoogle opt]$ ll
total 115168
-rw-r--r--. 1 root root 117929735 Oct 24 01:07 amazon-corretto-8-x64-linux-jdk.tar.gz
[ec2-user@samplegoogle opt]$ sudo tar -xzf amazon-corretto-8-x64-linux-jdk.tar.gz
[ec2-user@samplegoogle opt]$ ll
total 115168
drwxr-xr-x. 7 ansadmin ansadmin       188 Oct 21 07:07 amazon-corretto-8.272.10.3-linux-x64
-rw-r--r--. 1 root     root     117929735 Oct 24 01:07 amazon-corretto-8-x64-linux-jdk.tar.gz
[ec2-user@samplegoogle opt]$ sudo mv amazon-corretto-8.272.10.3-linux-x64 java-1.8
[ec2-user@samplegoogle opt]$
```

3. Change java permissions

sudo chmod -R 755 java-1.8

```
[ec2-user@samplegoogle opt]$ ll
total 115168
-rw-r--r--. 1 root     root     117929735 Oct 24 01:07 amazon-corretto-8-x64-linux-jdk.tar.gz
drwxr-xr-x. 7 ansadmin ansadmin       188 Oct 21 07:07 java-1.8
[ec2-user@samplegoogle opt]$ sudo chmod -R 755 java-1.8
[ec2-user@samplegoogle opt]$ ll
total 115168
-rw-r--r--. 1 root     root     117929735 Oct 24 01:07 amazon-corretto-8-x64-linux-jdk.tar.gz
drwxr-xr-x. 7 ansadmin ansadmin       188 Oct 21 07:07 java-1.8
[ec2-user@samplegoogle opt]$
```

Temporary setting (With in a session)

1. Java binaries path

/opt/java-1.8/bin

```
[ec2-user@samplegoogle bin]$ pwd
/opt/java-1.8/bin
[ec2-user@samplegoogle bin]$ ls -lrt java*
-rwxr-xr-x. 1 ansadmin ansadmin 2806 Oct 21 07:07 java-rmi.cgi
-rwxr-xr-x. 1 ansadmin ansadmin 2293 Oct 21 07:07 javapackager
-rwxr-xr-x. 1 ansadmin ansadmin 8856 Oct 21 07:07 javap
-rwxr-xr-x. 1 ansadmin ansadmin 8856 Oct 21 07:07 javah
-rwxr-xr-x. 1 ansadmin ansadmin 2293 Oct 21 07:07 javafxpackager
-rwxr-xr-x. 1 ansadmin ansadmin 8856 Oct 21 07:07 javadoc
-rwxr-xr-x. 1 ansadmin ansadmin 8856 Oct 21 07:07 javac
-rwxr-xr-x. 1 ansadmin ansadmin 8720 Oct 21 07:07 java
[ec2-user@samplegoogle bin]$
[ec2-user@samplegoogle bin]$
```

2. Verify the Java version

java –version

```
[ec2-user@samplegoogle bin]$ java -version
-bash: java: command not found
[ec2-user@samplegoogle bin]$
```

3. Setup java path and verify the java version

```
# PATH=$PATH:/opt/java-1.8/bin
# export PATH
# echo $PATH
# java -version
```

```
[ec2-user@samplegoogle bin]$ PATH=$PATH:/opt/java-1.8/bin
[ec2-user@samplegoogle bin]$ export PATH
[ec2-user@samplegoogle bin]$ echo $PATH
/home/ec2-user/.local/bin:/home/ec2-user/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/opt/java-1.8/bin
[ec2-user@samplegoogle bin]$ java -version
openjdk version "1.8.0_272"
OpenJDK Runtime Environment Corretto-8.272.10.3 (build 1.8.0_272-b10)
OpenJDK 64-Bit Server VM Corretto-8.272.10.3 (build 25.272-b10, mixed mode)
```

4. Exit the session and try to login again verify the java version

[ec2-user@samplegoogle ~]$ java -version
-bash: java: command not found
[ec2-user@samplegoogle ~]$

```
[ec2-user@samplegoogle bin]$ java -version
openjdk version "1.8.0_272"
OpenJDK Runtime Environment Corretto-8.272.10.3 (build 1.8.0_272-b10)
OpenJDK 64-Bit Server VM Corretto-8.272.10.3 (build 25.272-b10, mixed mode)
[ec2-user@samplegoogle bin]$ exit
logout


Session stopped
    - Press <return> to exit tab
    - Press R to restart session
    - Press S to save terminal output to file
Authenticating with public key "Imported-Openssh-Key: C:\Users\ppred\OneDrive\Desktop\awsdevops.pem"

                    • MobaXterm 10.9 •
            (SSH client, X-server and networking tools)

  ➤ SSH session to ec2-user@18.188.154.67
    • SSH compression : ✔
    • SSH-browser     : ✔
    • X11-forwarding  : ✘  (disabled or not supported by server)
    • DISPLAY         : 192.168.43.15:0.0

  ➤ For more info, ctrl+click on help or visit our website

This system is not registered to Red Hat Insights. See https://cloud.redhat.com/
To register this system, run: insights-client --register

Last login: Sat Oct 24 13:57:06 2020 from 157.49.166.91
[ec2-user@samplegoogle ~]$ java -version
-bash: java: command not found
[ec2-user@samplegoogle ~]$
```

Setting Path permanently in user level(ec2-user)

1. Open the .profile file and add the java path

# cd
# ls –a
# vi .profile
PATH=$PATH:/opt/java-1.8/bin
export PATH

```
[ec2-user@samplegoogle ~]$ cd
[ec2-user@samplegoogle ~]$ ls -a
.                             .config            folder1          sample_06-07.log     sample.txt
..                            devops_batch       folder2          sample_0629.log      softlink.txt
700                           devops.txt         hello.txt        sample_20050738.log  softppreddy
apache-maven-3.6.3-bin.tar.gz dir                .lesshst         sample_20050752.log  .ssh
apache-maven-3.6.3-bin.tar.gz.1 display          locate-demo.txt  sample_20050910.log  test
.bash_history                 file1.txt          logs             sample_20051312.log  .wget-hsts
.bash_logout                  file2.txt          mailbox          sample_50.log
.bash_profile                 findmoduledemo.txt perm.txt         sample_folder
.bashrc                       FINDmoduledemo.txt ppreddy          sample_script.sh
[ec2-user@samplegoogle ~]$ vi .bash_profile
```

2. Exit and try to login again
3. Verify the java version

java -version

```
[ec2-user@samplegoogle ~]$ vi .bash_profile
[ec2-user@samplegoogle ~]$ exit
logout


Session stopped
    - Press <return> to exit tab
    - Press R to restart session
    - Press S to save terminal output to file
Authenticating with public key "Imported-Openssh-Key: C:\Users\ppred\OneDrive\Desktop\awsdevops.pem"

                          • MobaXterm 10.9 •
                (SSH client, X-server and networking tools)

  ➤ SSH session to ec2-user@18.188.154.67
    • SSH compression : ✔
    • SSH-browser     : ✔
    • X11-forwarding  : ✘  (disabled or not supported by server)
    • DISPLAY         : 192.168.43.15:0.0

  ➤ For more info, ctrl+click on help or visit our website

This system is not registered to Red Hat Insights. See https://cloud.redhat.com/
To register this system, run: insights-client --register

Last login: Sat Oct 24 14:15:17 2020 from 157.49.166.91
[ec2-user@samplegoogle ~]$ java -version
openjdk version "1.8.0_272"
OpenJDK Runtime Environment Corretto-8.272.10.3 (build 1.8.0_272-b10)
OpenJDK 64-Bit Server VM Corretto-8.272.10.3 (build 25.272-b10, mixed mode)
[ec2-user@samplegoogle ~]$
```

Setting the path global/sudo level

1. Verify the java version

java –version

```
[ec2-user@samplegoogle ~]$ java -version
-bash: java: command not found
[ec2-user@samplegoogle ~]$
```

2. Switch to sudo user

#sudo su

#whoami

```
[ec2-user@samplegoogle ~]$ java -version
-bash: java: command not found
[ec2-user@samplegoogle ~]$ sudo su
[root@samplegoogle ec2-user]# whoami
root
[root@samplegoogle ec2-user]#
```

3. Moved to profile.d and add the paths file and exit

cd /etc/profile.d

vi paths.sh

```
PATH=$PATH:/opt/java-1.8/bin
export PATH
```

#exit

```
[root@samplegoogle profile.d]# exit
exit
[ec2-user@samplegoogle ~]$ exit
logout


Session stopped
    - Press <return> to exit tab
    - Press R to restart session
    - Press S to save terminal output to file
```

4. Verify the java version

java -version

```
[root@samplegoogle profile.d]# exit
exit
[ec2-user@samplegoogle ~]$ exit
logout


Session stopped
    - Press <return> to exit tab
    - Press R to restart session
    - Press S to save terminal output to file
Authenticating with public key "Imported-Openssh-Key: C:\Users\ppred\OneDrive\Desktop\awsdevops.pem"

                          • MobaXterm 10.9 •
                (SSH client, X-server and networking tools)

  ➤ SSH session to ec2-user@18.188.154.67
    • SSH compression : ✔
    • SSH-browser     : ✔
    • X11-forwarding  : ✘  (disabled or not supported by server)
    • DISPLAY         : 192.168.43.15:0.0

  ➤ For more info, ctrl+click on help or visit our website

This system is not registered to Red Hat Insights. See https://cloud.redhat.com/
To register this system, run: insights-client --register

Last login: Sat Oct 24 14:27:39 2020 from 157.49.166.91
[ec2-user@samplegoogle ~]$ java -version
openjdk version "1.8.0_272"
OpenJDK Runtime Environment Corretto-8.272.10.3 (build 1.8.0_272-b10)
OpenJDK 64-Bit Server VM Corretto-8.272.10.3 (build 25.272-b10, mixed mode)
[ec2-user@samplegoogle ~]$
```

5. Create the new user javauser

```
[root@samplegoogle ec2-user]# useradd javauser
[root@samplegoogle ec2-user]# tail /etc/passwd
ansadmin:x:1001:1001::/home/ansadmin:/bin/bash
saslauth:x:993:76:Saslauthd user:/run/saslauthd:/sbin/nologin
mailnull:x:47:47::/var/spool/mqueue:/sbin/nologin
smmsp:x:51:51::/var/spool/mqueue:/sbin/nologin
awsuser1:x:1002:1002::/home/awsuser1:/bin/bash
awsuser2:x:1003:1004::/home/awsuser2:/bin/bash
awsuser3:x:1004:1005::/home/awsuser3:/bin/bash
awsadmin:x:1005:1006::/home/awsadmin:/bin/bash
dummy:x:1006:1007::/home/dummy:/bin/bash
javauser:x:1007:1008::/home/javauser:/bin/bash
[root@samplegoogle ec2-user]#
```

6. Switch to the javauser and verify the java version.

su javauser

java –version

```
[root@samplegoogle ec2-user]# su javauser
[javauser@samplegoogle ec2-user]$ cd
[javauser@samplegoogle ~]$ pwd
/home/javauser
[javauser@samplegoogle ~]$ java -version
openjdk version "1.8.0_272"
OpenJDK Runtime Environment Corretto-8.272.10.3 (build 1.8.0_272-b10)
OpenJDK 64-Bit Server VM Corretto-8.272.10.3 (build 25.272-b10, mixed mode)
[javauser@samplegoogle ~]$
```