**LINUX COMMANDS**

1**.awk** - awk is mostly used for pattern scanning and processing. It searches one or more files to see if they contain lines that matches with the specified patterns and then perform the associated actions.

 Syntax: awk options 'selection \_criteria {action }' input-file >    output-file

 Example: Consider the below text file

  $ cat > employee.txt

     ajay manager account 45000

     sunil clerk account 25000

     varun manager sales 50000

     1.By using awk command we can print every line of data.

     $ awk '{print}' employee.txt

     Output:

     ajay manager account 45000

     sunil clerk account 25000

     varun manager sales 50000

      2.By using awk command we can print the lines which matches with the given pattern

      $ awk '/manager/ {print}' employee.txt

      Output:

       ajay manager account 45000

       varun manager sales 5000

    Built-In Variables In Awk

          Awk’s built-in variables include the field variables—$1, $2, $3, and so on ($0 is the      entire line) — that break a line of text into individual words or pieces called fields.

1.NR: NR command keeps a current count of the number of input records. Remember that records are usually lines. Awk command performs the pattern/action statements once for each record in a file.

      $ awk '{print NR,$0}' employee.txt

      Output:

1. ajay manager account 45000

  2    sunil clerk account 25000

  3    varun manager sales 50000

2.NF: NF command keeps a count of the number of fields within the current input record.

    $ awk '{print $1,$NF}' employee.txt

   Output:

     ajay 45000

     sunil 25000

     varun 5000

2.**grep** -  processes text line by line, and prints any lines which match a specified pattern. The grep command is used for searching the text from the file according to the regular expression.

         -i: Ignore case when searching

         -r: search recursively in directories

         -v: displays lines that do not match the pattern

**-c**: This prints only a count of the lines that match a pattern

**-l:** Displays list of a filenames only.

        Syntax: grep [options]  pattern [files]

        Example:

       $cat > sample.txt

       unix is great os. unix was developed in Bell labs.

       learn operating system.

       Unix linux which one you choose.

       uNix is easy to learn.unix is a multiuser os.

1. Case insensitive search:The -i option enables to search for a string case insensitively in the given file. It matches the words like “UNIX”, “Unix”, “unix”.

$grep -i "UNix" sample.txt

  Output:

  unix is great os. unix was developed in Bell labs.

   Unix linux which one you choose.

   uNix is easy to learn.unix is a multiuser os.

2.Displaying the count of number of matches**:** We can find the number of lines that matches the given string/pattern

$grep -c "unix" sample.txt

Output:2

3.**sed** - SED command in UNIX stands for stream editor and it can perform lots of functions on file like searching, find and replace, insertion or deletion.

Syntax:sed OPTIONS... [SCRIPT] [INPUTFILE...]

Example: consider the below text file

         $ Cat sample.txt

         unix is great os. unix is opensource. unix is free os.

         learn operating system.

         unix linux which one you choose.

1.Replacing or substituting string **:** Sed command is mostly used to replace the text in a file. The below simple sed command replaces the word “unix” with “linux” in the file.

$sed 's/unix/linux/' geekfile.txt

Output:

linux is great os. unix is opensource. unix is free os.

learn operating system.

linux linux which one you choose.

2.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/unix/linux/2' sample.txt

Output:

unix is great os. linux is opensource. unix is free os.

learn operating system.

unix linux which one you choose.

4**.Cut** - 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**.**

Syntax:

cut OPTION... [FILE]...

Example:

Consider state.txt

Andhra Pradesh

Arunachal Pradesh

Assam

Bihar

Chhattisgarh

1**.**-b(byte): To extract the specific bytes, you need to follow -b option with the list of byte numbers separated by comma.

$ cut -b 1,2,3 state.txt

And

Aru

Ass

Bih

Chh

$ cut -b 1- state.txt

Andhra Pradesh

Arunachal Pradesh

Assam

Bihar

Chhattisgarh

2**.**-c (column): To cut by character use the -c option. This selects the characters given to the -c option. Tabs and backspacesare treated as a character.

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

nr

rah

sm

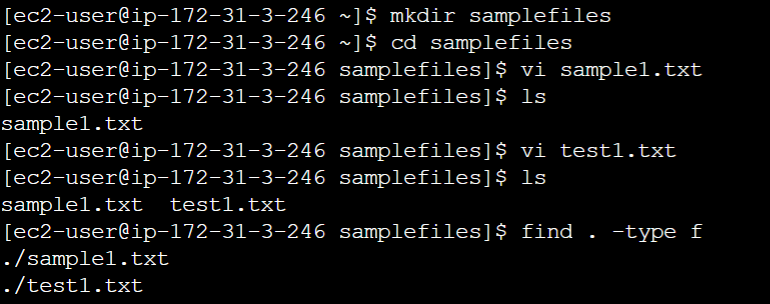
ir

hti

5.**Find** - The find command is used to find a particular file within a directory. It also supports various options to find a file such as byname, by type, by date.

  -name: Searches for files and directories with a specific name.

  -type: Searches for files or directories of a specific type(eg:file,directory)



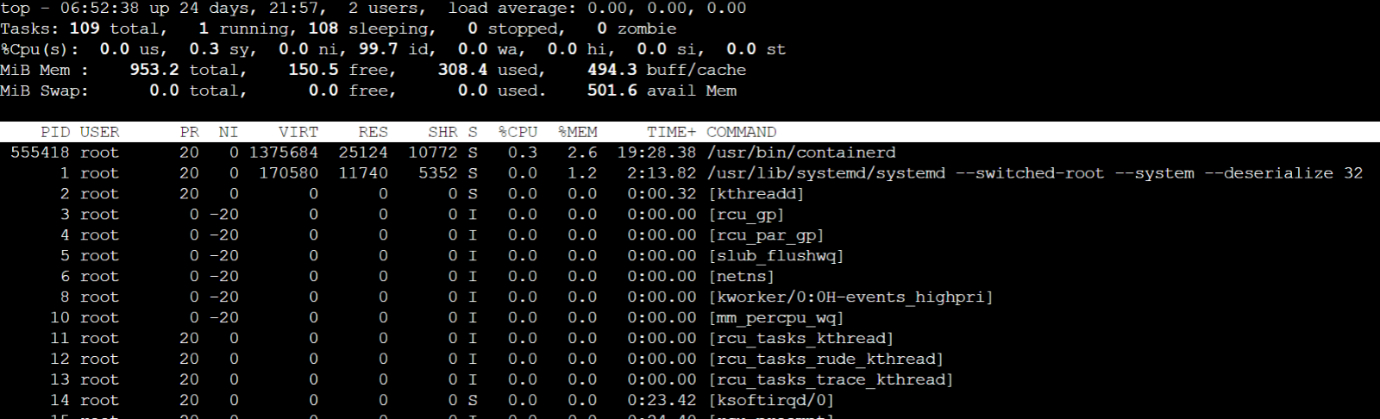
6.**top**: Displays information about running process

-c: shows the full command line for each process.

-k: sends a signal to a process to terminate it.

-u: filters processes by user

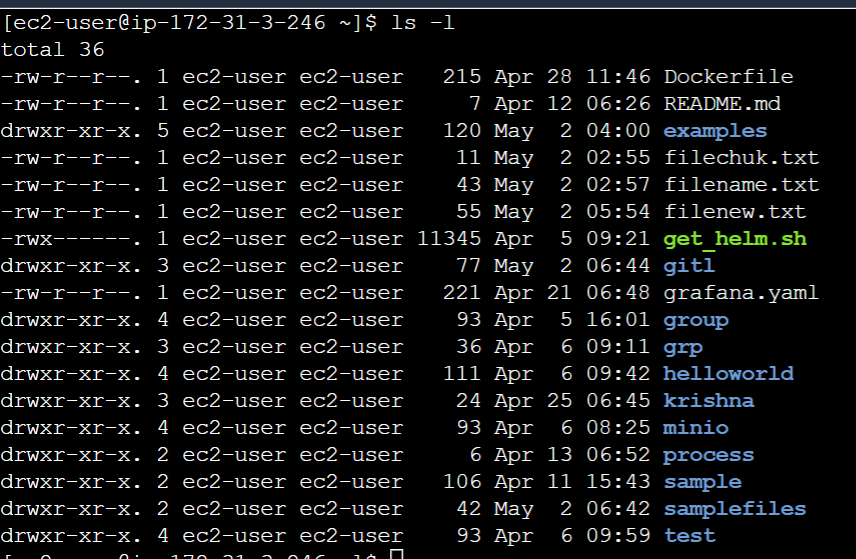
-p: filters processes by process ID



7.**ls** – lists the contents of a directory

  -a: Displays all files and directories ,including hidden files

  -l: Displays the contents in a long format, including  permissions, ownership,size, and modification date.



8**.rm**: Removes files or directories

-r: Recursively removes directories and their contents.

 –f: forcefully removes directories and their contents.

Syntax: rm filename –r

9.**cp**: copies files or directories.

 -r: Recursively copies directories and its contents

 -f: forcefully overwrite of destination files without prompting for confirmation.

Syntax: cp -R Src\_directory Dest\_directory

10.**mv**: Moves or renames files or directories.

       -i: prompts for confirmation before overwriting existing files.

       -u: only overwrites files that are newer than the destination file.

    $ ls

b.txt  c.txt  d.txt  geek.txt

$ cat geek.txt

India

$ cat b.txt

geeksforgeeks

$ mv -i geek.txt b.txt

mv: overwrite 'b.txt'? y

$ ls

b.txt  c.txt  d.txt

$ cat b.txt

India

11.**tar**: Archives files into a single file

-c: creates a new archive

-x: Extracts files from archive

-z: compresses the archieve using gzip

-f: specifies the filename of the archieve

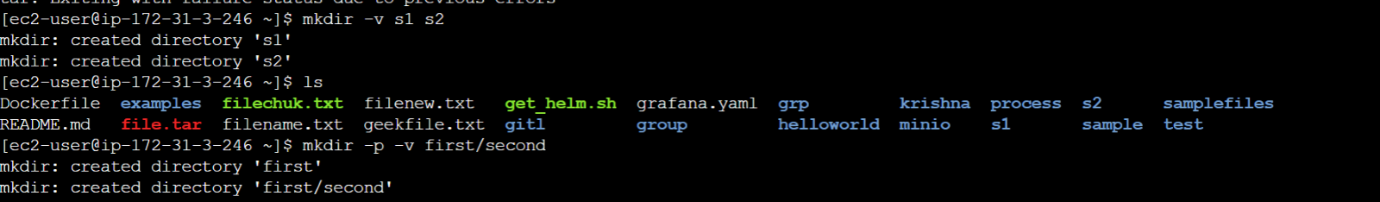
Syntax:To extract the files from archieve

 tar xvf file.tar

12**.mkdir**: mkdircommand in Linux allows the user to create directories (also referred to as folders in some operating systems). This command can create multiple directories at once as well as set the permissions for the directories.

-v:  It displays a message for every directory created.

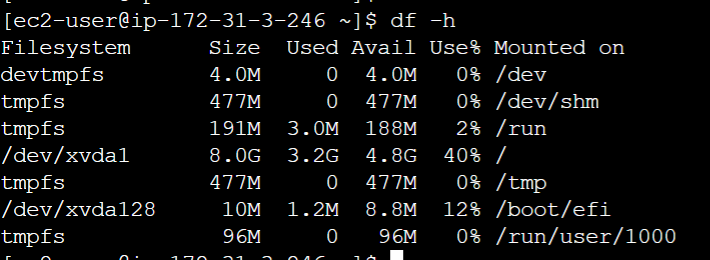
-p: A flag which enables the command to create parent directories as necessary. If the directories exist, no error is specified.



13**.df**: It is used to display information about file system disk space usage.

-h: this flag displays the disk space in human-readable format,such as GB,MB,KB.

-T: this flag displays the file system type along with the disk space usage.

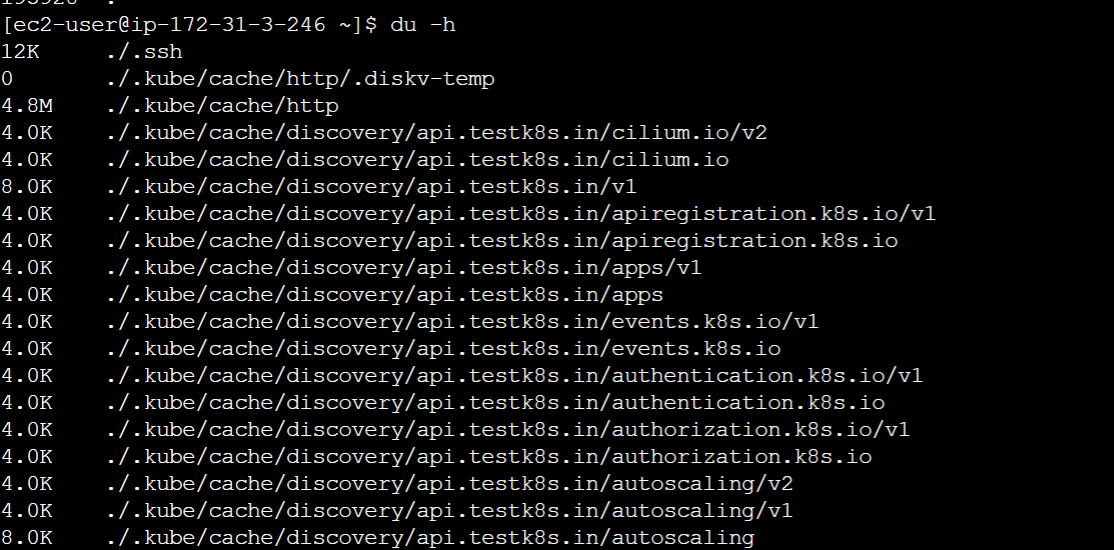


14.**du**: It is used to estimate the file space usage of a directory and its subdirectories.

-h: it displays the disk space in human-readable format,such as GB,MB,KB.

-s: this flag displays the total disk space usage of the directory,without listing the disk space usage of each file and subdirectory.

-c: this flag displays the grand total of the disk space usage of all specified directories



15.**chmod**: changes the permissions of a file or directory.

-u: changes the permissions for the owner of the file.

-g: changes the permissions for the group associated with the file.

-o: changes the permissions for others

-a: changes the permissions for all users.

-‘+’: adds  a permission

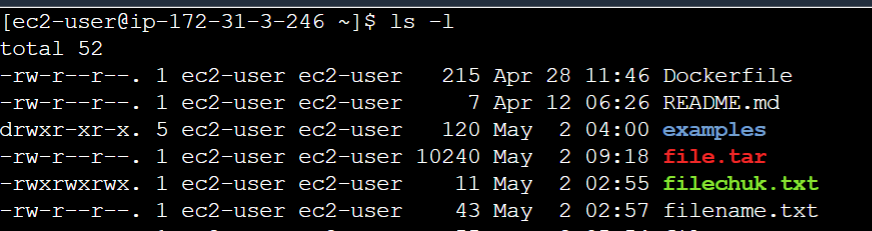
-‘\_ : removes a permission

‘=’: sets the permission to the specified value.

* Read, write and execute permissions to the file owner:

chmod u+rwx [file\_name]

To view the current permissions of a file or directory.



16.usermod: usermod command or modify user is a command in Linux that is used to change the properties of a user in Linux through the command line.

To change the username of the existing user.

Command: sudo usermod –l newusername oldusername

Here –l flag is used to change the username of the user ‘oldusername’ to ‘newusername’

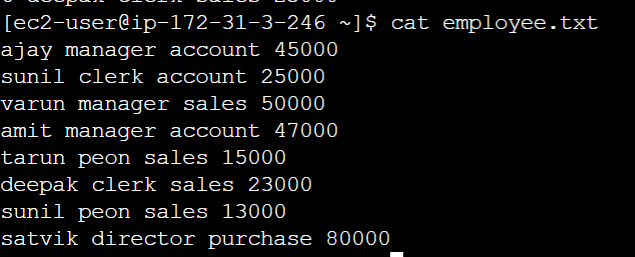
17.ssh command: Connects to a remote server using SSH

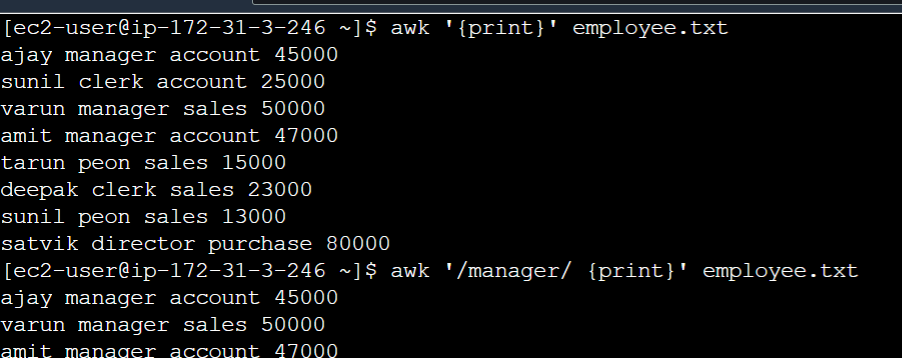
     -p: specifies the port to connect to.

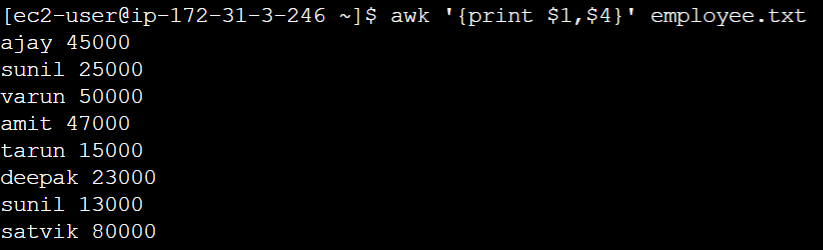
     -i: specifies the identity file (i.e private key) to use for authentication

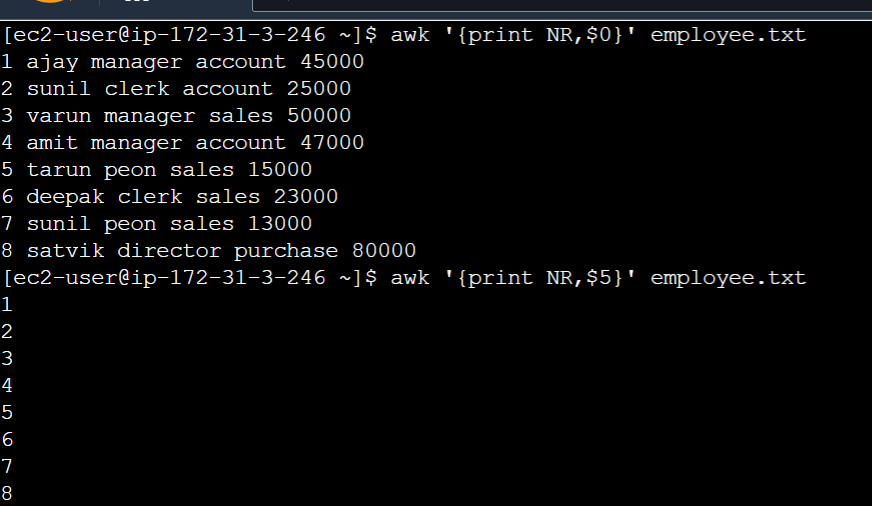
      Command: ssh -p 2222 user@host

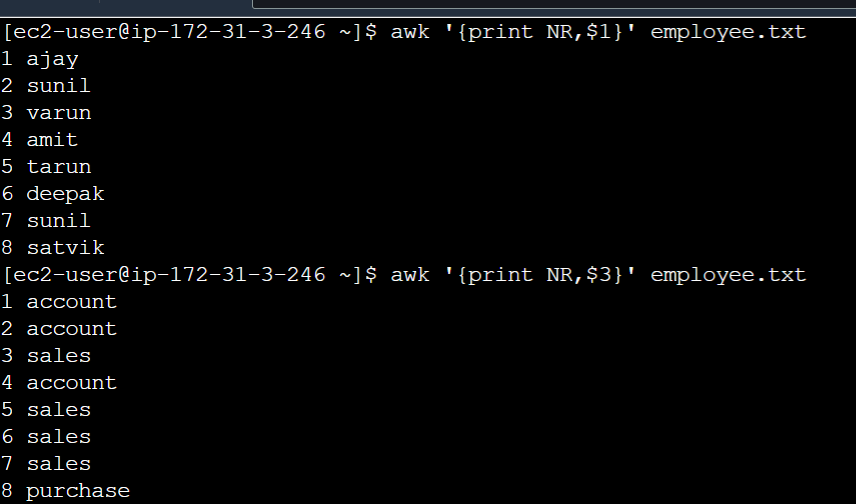
1.awk command:

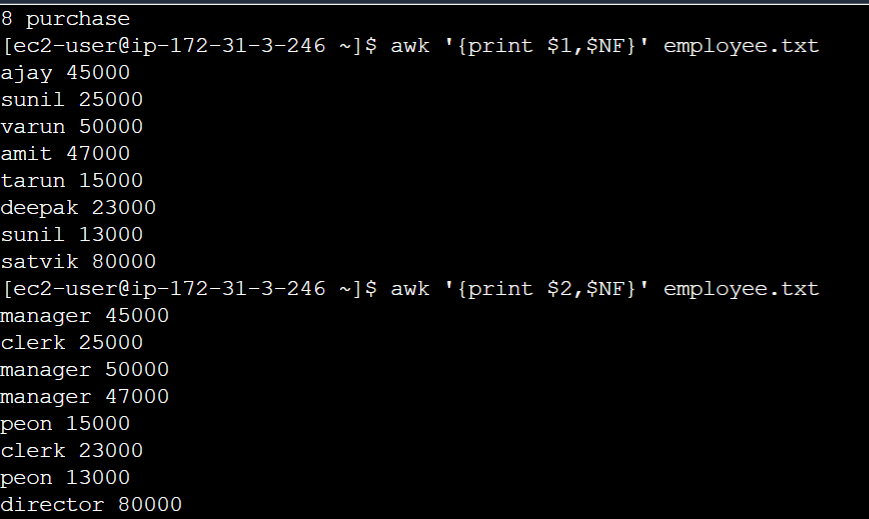


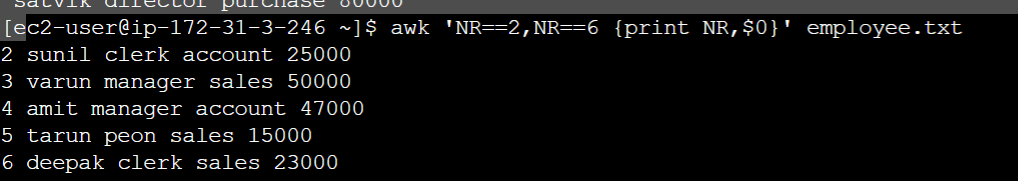




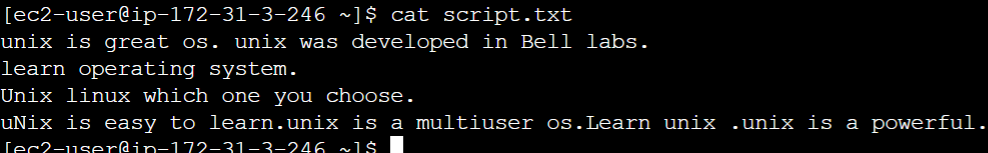


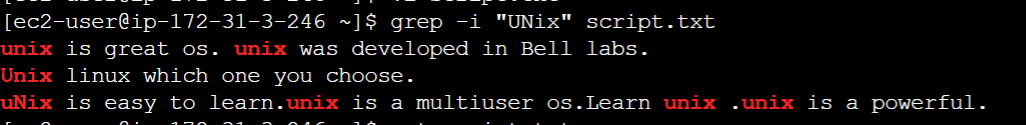


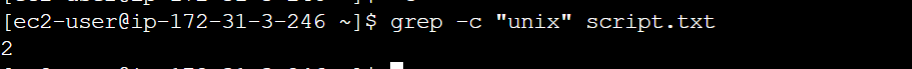


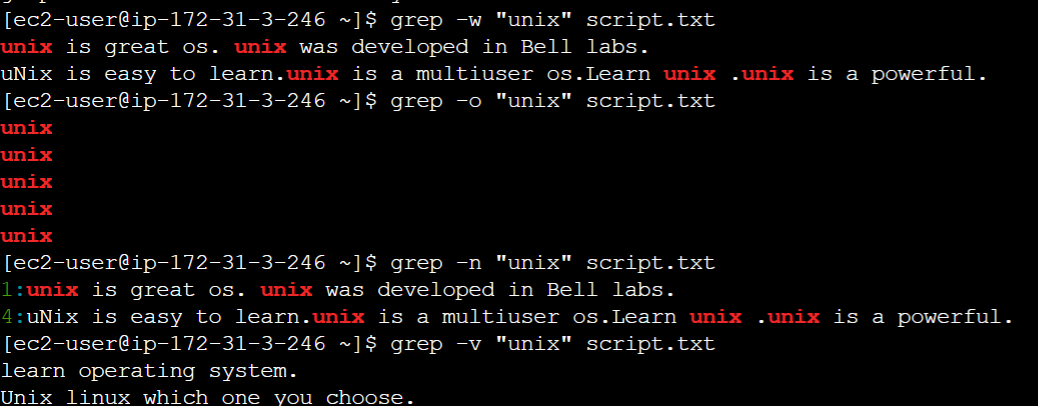


**Grep:**

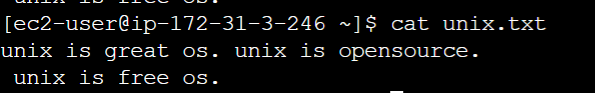


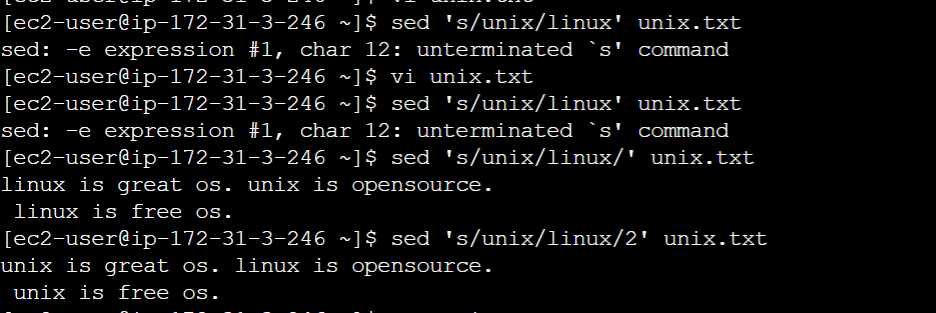




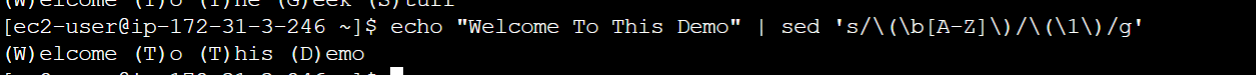


3.sed:

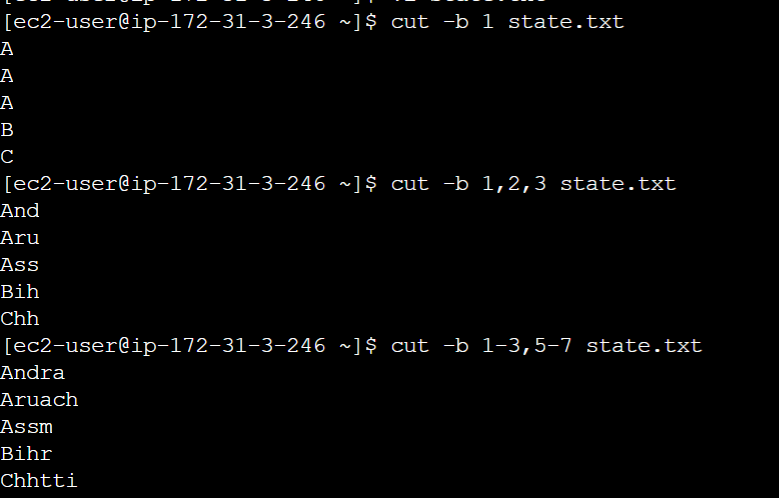


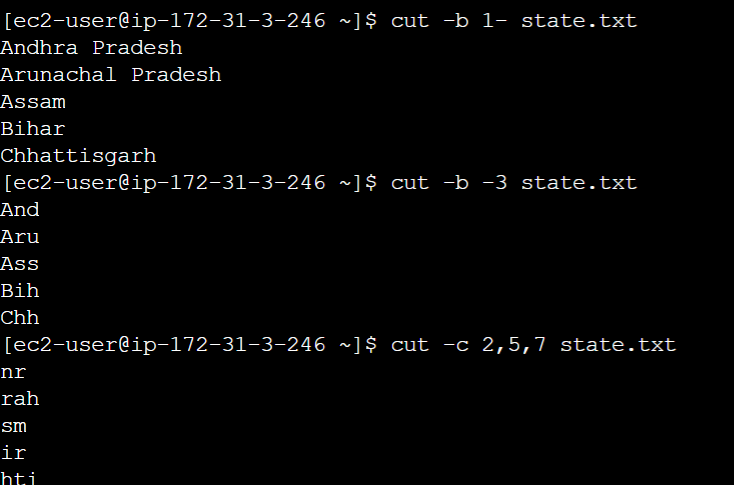


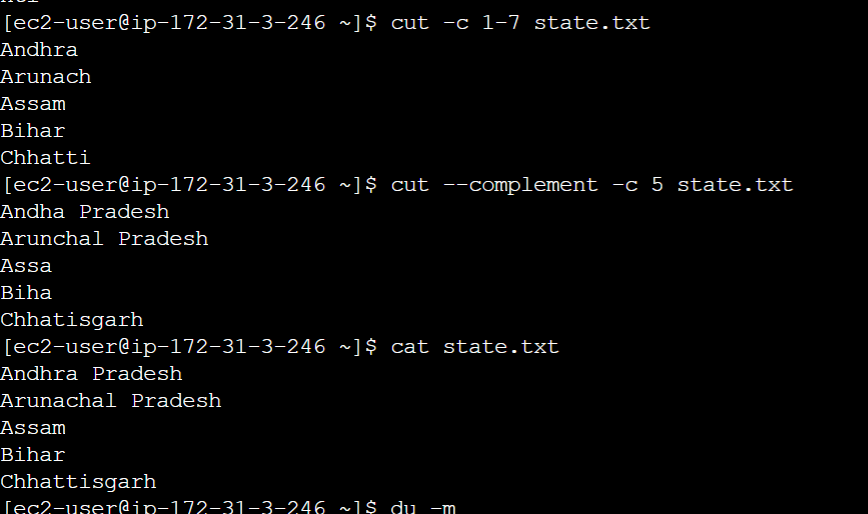
**Parenthesize first character of each word :**This sed example prints the first character of every word in parenthesis.



4.cut command:







**HOW TO CREATE DIR WITHIN DIR**

**mkdir -p demo/demo1/demo2/demo.txt**

**CHECK THE CURRENT WORKING PROCESS:- PS**

**DISK FREE SPACE- DF-H  
DISK USAGE- DU**