Module 4: Linux Fundamentals for DevOps

Linux Commands

edureka!

Basic Linux Commands

Linux provides a **CLI** (Command Line Interface) to communicate with the OS. Here are the most basic of the Linux Commands.

To open a terminal, click ***Show Applications*** option in the left-hand side bar, and browse and open ***Terminal.***
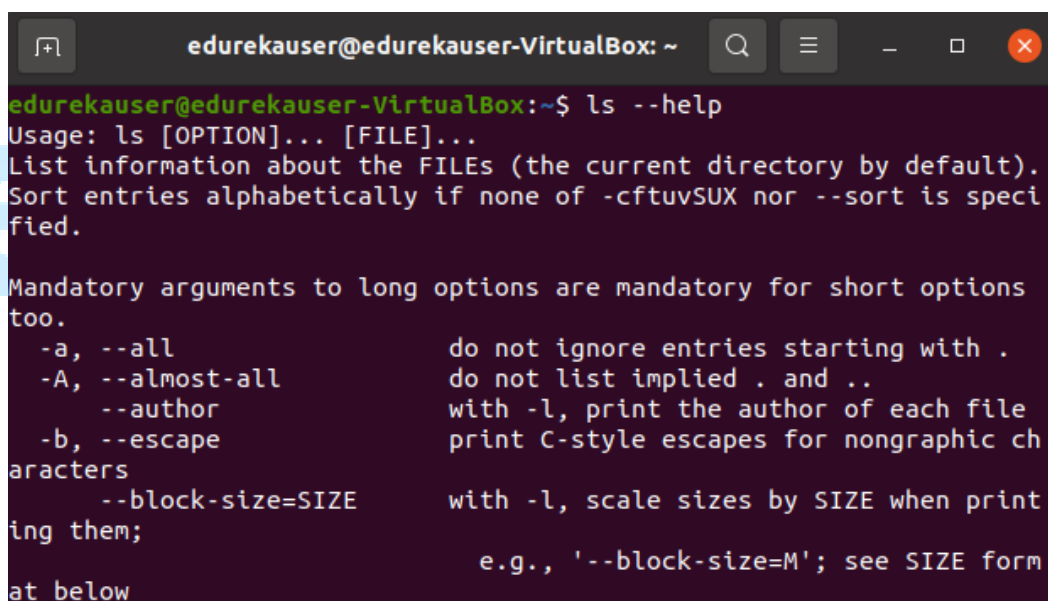
Note: Refer the document ***Linux VM Setup Guide*** under Setup Guides on LMS to install and run a Linux Virtual Machine on your local machine.

1. help

Displays the information on the usage of a command

syntax:

$ <command> --help

2. ls

Lists all the contents in the current working directory.

syntax:

$ ls <flag>

```
edurekauser@edurekauser-VirtualBox:~$ ls
Desktop      Downloads  Pictures  Templates
Documents  Music      Public    Videos
```

Command	Explanation
ls <path name>	By specifying the path after ls, the content in that path will be displayed
ls –l	Using 'l' flag, lists all the contents along with its owner settings, permissions & time stamp (long format)
ls –a	Using 'a' flag, lists all the hidden contents in the specified directory
ls –author	Using '–author' flag, lists the contents in the specified directory along with its owner
ls –s	Using 's' flag, sorts and lists all the contents in the specified directory by size
*ls *.html*	Using '*' flag, lists only the contents in the directory of a particular format
ls –lS > file.txt	Using '>' flag, copies the result of preceding ls command into a text file

3. mkdir

Creates a new directory.

syntax:

$ mkdir <flag> {directoryname} /pathname/

```
edurekauser@edurekauser-VirtualBox:~$ mkdir newdirectory
edurekauser@edurekauser-VirtualBox:~$ ls
Desktop     Downloads   newdirectory  Public     Videos
Documents   Music       Pictures      Templates
```

Command	Explanation
mkdir -p	Creates both a new parent directory and a sub-directory
mkdir – p <filename1>/{f1,f2,f3}	This is used to create multiple subdirectories inside the new parent directory

4. touch

Creates a file or changes a file's timestamp.

syntax:

$ touch <flag> {filename}

```
edurekauser@edurekauser-VirtualBox:~$ touch testfile.txt
edurekauser@edurekauser-VirtualBox:~$ ls
Desktop     Downloads   newdirectory  Public     testfile.txt
Documents   Music       Pictures      Templates  Videos
```

5. mv

Moves files and directories from one directory to another. The file/directory once moved, is deleted from the working directory.

syntax:

$ mv <flag> {filename} /pathname/

```
edurekauser@edurekauser-VirtualBox:~$ mv testfile.txt /home/edurekau
ser/Public
edurekauser@edurekauser-VirtualBox:~$ ls /home/edurekauser/Public
testfile.txt
edurekauser@edurekauser-VirtualBox:~$ ls
Desktop     Downloads   newdirectory  Public     Videos
Documents   Music       Pictures      Templates
```
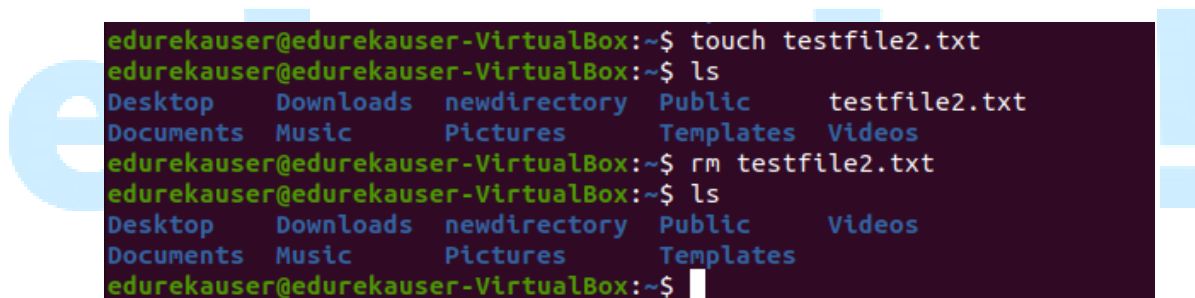
Command	Explanation
mv -i	Enters interactive mode; CLI asks before overwriting files
mv -u	Updates the destination file only when the source file is different from the destination file
mv -v	Verbose; Prints source and destination files

6. rm

Removes files from a directory. By default, the rm command does not remove directories. Once removed, the contents of a file cannot be recovered.

syntax:

$ rm <flag> {filename}

```
edurekauser@edurekauser-VirtualBox:~$ touch testfile2.txt
edurekauser@edurekauser-VirtualBox:~$ ls
Desktop     Downloads  newdirectory  Public     testfile2.txt
Documents   Music      Pictures      Templates  Videos
edurekauser@edurekauser-VirtualBox:~$ rm testfile2.txt
edurekauser@edurekauser-VirtualBox:~$ ls
Desktop     Downloads  newdirectory  Public     Videos
Documents   Music      Pictures      Templates
edurekauser@edurekauser-VirtualBox:~$
```

Command	Explanation
rm –r	Removes even non-empty directories.
rm –rp	Removes non-empty directories including parent and subdirectories.

7. cat

Used to read, modify, or concatenate text files. It also displays file contents.

syntax:

$ cat <flag> {filename}

```
edurekauser@edurekauser-VirtualBox:~$ cat cattestfile.txt
This is a file to test the concatenate command.
```

Command	Explanation
cat -b	This is used to add line numbers to non-blank lines
cat -n	This is used to add line numbers to all lines
cat -s	This is used to squeeze blank lines into one line
cat –e	Show $ at the end of line

8. tail

Displays the last few lines of a file.

syntax:

$ tail <flag> {filename}

```
edurekauser@edurekauser-VirtualBox:~$ tail cattestfile.txt
This is a file to test the concatenate command.
---
End of file.
```

9. cp

Copies files and directories. A copy of the file/directory copied, still remains in the working directory.

syntax:

$ cp <flag> {filename} /pathname/

```
edurekauser@edurekauser-VirtualBox:~$ cp cattestfile.txt /home/edurekauser/Publ
ic
edurekauser@edurekauser-VirtualBox:~$ ls /home/edurekauser/Public
cattestfile.txt   testfile.txt
edurekauser@edurekauser-VirtualBox:~$ ls
cattestfile.txt   Documents   Music        Pictures   Templates
Desktop           Downloads   newdirectory Public     Videos
edurekauser@edurekauser-VirtualBox:~$
```

Command	Explanation

cp -i	Enters interactive mode; CLI asks before overwriting files
cp -n	Does not overwrite the file
cp -u	Updates the destination file only when the source file is different from the destination file
cp -r	Recursive copy for copying directories; Copies even hidden files
cp -v	Verbose; Prints informative messages

10. pwd

Displays the current working directory of the terminal.

syntax:

$ pwd

```
edurekauser@edurekauser-VirtualBox:~$ pwd
/home/edurekauser
edurekauser@edurekauser-VirtualBox:~$
```

11. echo

Writes its arguments to standard output.

syntax:

$ echo "<text>"

```
edurekauser@edurekauser-VirtualBox:~$ echo "Hello Edureka Learner!"
Hello Edureka Learner!
edurekauser@edurekauser-VirtualBox:~$
```

12. clear

Clears the terminal screen. Contents will not actually be deleted in this case, only scrolled down. You can also clear the screen by pressing *Ctrl+L* on the keyboard.

syntax:

$ clear

13. whoami

Tells the current logged-in user.

syntax:

$ whoami

```
edurekauser@edurekauser-VirtualBox:~$ whoami
edurekauser
edurekauser@edurekauser-VirtualBox:~$
```

14. cd

Used to change the current working directory of the user.

syntax:

$ cd /pathname/

```
edurekauser@edurekauser-VirtualBox:~$ cd /home/edurekauser/Public
edurekauser@edurekauser-VirtualBox:~/Public$
```

Command	Explanation
cd ~	This command also changes the directory to home directory
cd /	Changes the directory to root directory
cd ..	Changes the directory to its parent directory
cd 'xx yy'	We specify the folder name in inverted commas because there is a space in the folder name

15. df

Shows information about file system.

syntax:

$ df <flag> <filename>

16. du

Shows disc usage of files recursively.

syntax:

$ du <flag> <filename>

ADVANCED FILE RELATED COMMANDS

1. grep

Searches for a particular string/word in a text file. This is similar to "Ctrl+F" but, executed via a CLI.

syntax:

$ grep <flag or element_to_search> {filename}

```
edurekauser@edurekauser-VirtualBox:~$ cat cattestfile.txt
This is a file to test the concatenate command.
---
End of file.
edurekauser@edurekauser-VirtualBox:~$ grep file cattestfile.txt
This is a file to test the concatenate command.
End of file.
edurekauser@edurekauser-VirtualBox:~$
```

Command	Explanation
grep -i	Returns the results for case insensitive strings
grep -n	Returns the matching strings along with their line number
grep -v	Returns the result of lines not matching the search string
grep -c	Returns the number of lines in which the results matched the search string

2. sed

A stream editor used to perform basic text transformations on an input stream (an input file or a input from a pipeline).

syntax:

$ sed <flag> {filename}

```
edurekauser@edurekauser-VirtualBox:~$ cat cattestfile.txt
This is a file to test the concatenate command.
---
End of file.
edurekauser@edurekauser-VirtualBox:~$ sed 's/concatenate/sed/' cattestfile.txt
> output.txt
edurekauser@edurekauser-VirtualBox:~$ cat output.txt
This is a file to test the sed command.
---
End of file.
edurekauser@edurekauser-VirtualBox:~$
```

Command	Explanation
sed -f	Adds the contents of script file to the commands for execution
sed -e	Adds the script to the command
sed -r	Allows to use regular expressions in the script
sed -z	Separates lines by NULL characters
sed -n	Prevents automatic printing of pattern space
sed -s	Considers files as separate rather than a single stream

3. **history**

Displays or manipulates the history list of commands.

```
edurekauser@edurekauser-VirtualBox:~$ history 5
   47  cat output.txt
   48  sed --help
   49  man sed
   50  history --help
   51  history 5
edurekauser@edurekauser-VirtualBox:~$ ls
cattestfile.txt  Documents  Music        output.txt  Public     Videos
Desktop          Downloads  newdirectory Pictures    Templates
edurekauser@edurekauser-VirtualBox:~$ history 5
   49  man sed
   50  history --help
   51  history 5
   52  ls
   53  history 5
edurekauser@edurekauser-VirtualBox:~$
```

4. sort

Sorts the results of a search either alphabetically or numerically. Files, file contents and directories can be sorted using this command.

syntax:

$ sort <flag> {filename}

```
edurekauser@edurekauser-VirtualBox:~$ cat sorttestfile.txt
ls
pwd
mv
rm
cat
cd
edurekauser@edurekauser-VirtualBox:~$ sort sorttestfile.txt
cat
cd
ls
mv
pwd
rm
edurekauser@edurekauser-VirtualBox:~$
```

Command	Explanation
sort -r	the flag returns the results in reverse order;
sort -f	the flag does case insensitive sorting
sort -n	the flag returns the results as per numerical order

User Management in Linux

1. su

Switches to root-user so that superuser permissions can be used to execute commands.

syntax:

$ su

2. su <username>

Switches to a different user whose name is passed as the argument.

syntax:

$ su <username>

3. sudo

Executes only that command with root/superuser privileges.

syntax:

$ sudo <command>

```
edurekauser@edurekauser-VirtualBox:~$ cat /etc/shadow
cat: /etc/shadow: Permission denied
edurekauser@edurekauser-VirtualBox:~$ sudo cat /etc/shadow
[sudo] password for edurekauser:
root:!:18614:0:99999:7:::
daemon:*:18474:0:99999:7:::
bin:*:18474:0:99999:7:::
sys:*:18474:0:99999:7:::
sync:*:18474:0:99999:7:::
games:*:18474:0:99999:7:::
```

```
edurekauser@edurekauser-VirtualBox:~$ tail -2 /etc/passwd
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
vboxadd:x:998:1::/var/run/vboxadd:/bin/false
edurekauser@edurekauser-VirtualBox:~$ sudo useradd student
edurekauser@edurekauser-VirtualBox:~$ tail -2 /etc/passwd
vboxadd:x:998:1::/var/run/vboxadd:/bin/false
student:x:1001:1001::/home/student:/bin/sh
edurekauser@edurekauser-VirtualBox:~$
```

```
edurekauser@edurekauser-VirtualBox:~$ tail -2 /etc/passwd
vboxadd:x:998:1::/var/run/vboxadd:/bin/false
student:x:1001:1001::/home/student:/bin/sh
edurekauser@edurekauser-VirtualBox:~$ sudo userdel student
edurekauser@edurekauser-VirtualBox:~$ tail -2 /etc/passwd
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
vboxadd:x:998:1::/var/run/vboxadd:/bin/false
edurekauser@edurekauser-VirtualBox:~$
```

Command	Explanation
sudo useradd <username>	Adding a new user
sudo passwd <username>	Setting a password for the new user
sudo userdel <username>	Deleting the user
sudo groupadd <groupname>	Adding a new group
sudo groupdel <groupname>	Deleting the group
sudo usermod -g <groupname> <username>	Adding a user to a primary group

Ownership and Permission

To view permissions of existing files-

1. chown

Used to change the user/owner of a file.

syntax:

$ chown <username> {filename}

2. chmod

Used to change the access permissions of files and directories. Consider the example below.

syntax:

$ chmod <permissions of user,group,others> {filename}

The permissions associated with each digit are as follows-

Number	read	write	execute
0	–	–	–
1	–	–	yes
2	–	yes	–
3	–	yes	yes
4	yes	–	–
5	yes	–	yes
6	yes	yes	–
7	yes	yes	yes