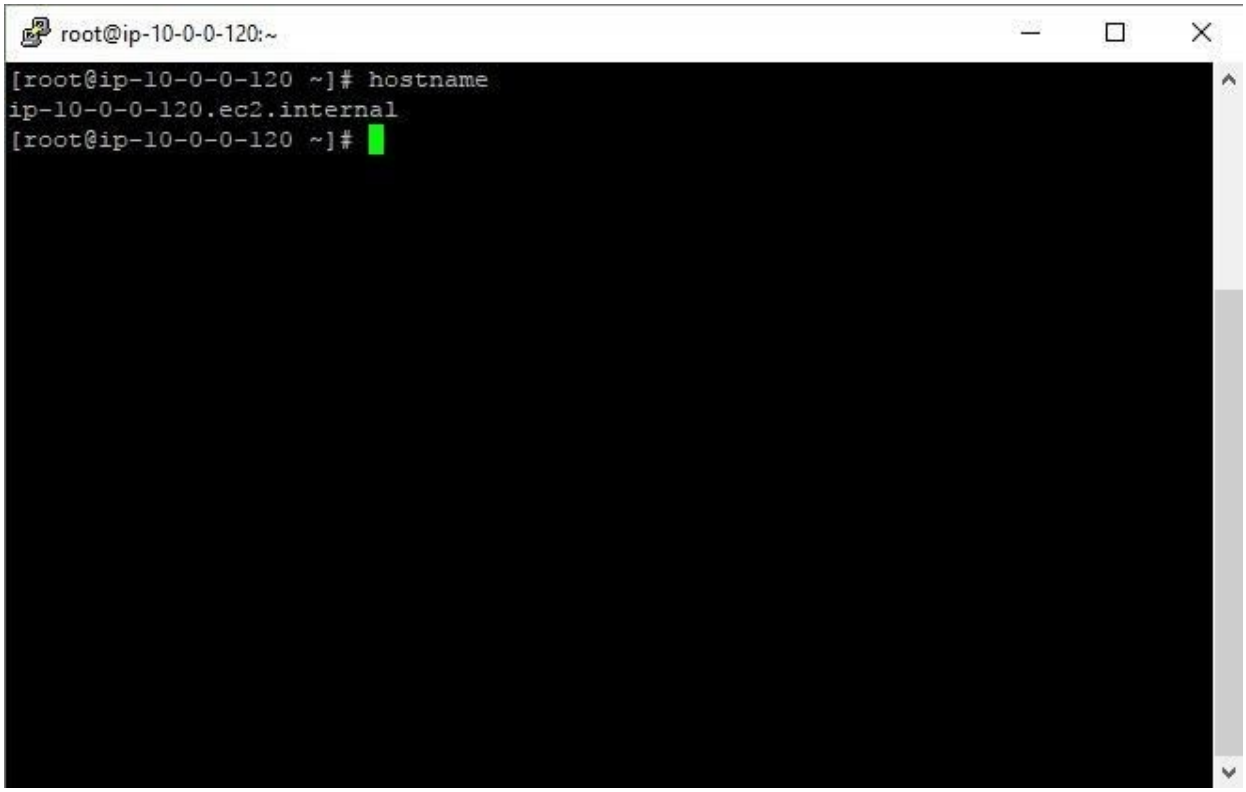
BASIC LINUX COMMANDS

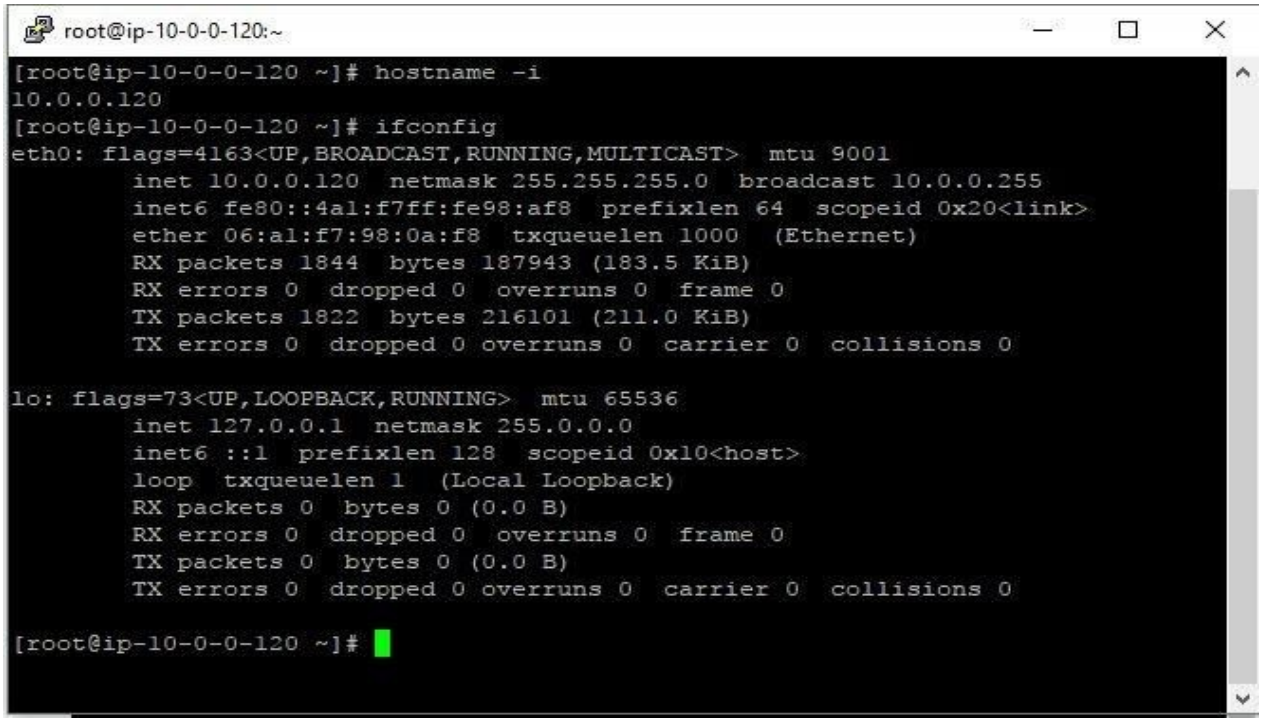
In this file, you will learn about various Linux commands and understand their usage to execute the instructions given in the course.

Commands to check machine configuration:

1. The hostname command: It is used to show or set the system's hostname. It can also be used to display the system's DNS name, or to display or set the NIS (Network Information Services) domain name. In many cases, you will use this command to display the system's hostname.
    a. To display a hostname: **hostname**



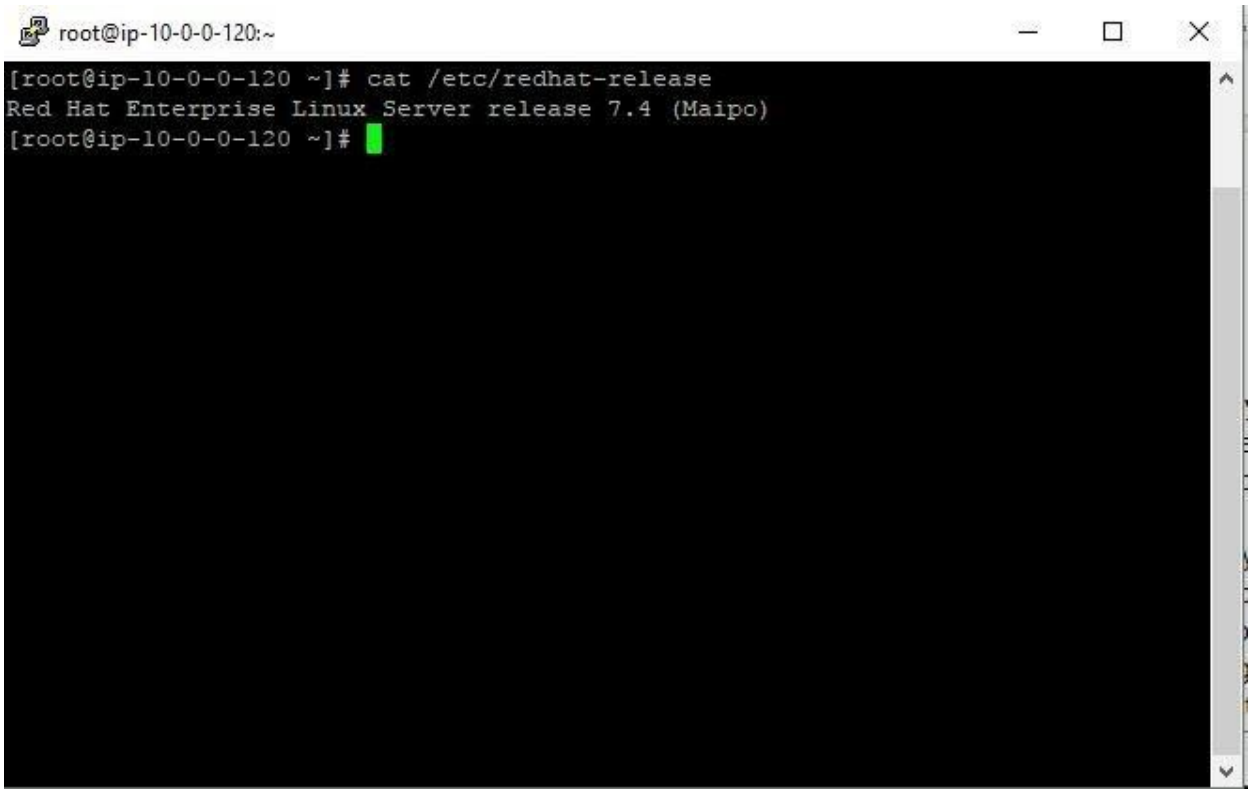
b. To display an IP address: **hostname -i** or **ifconfig**



c. To display the domain name: **hostname -d**

d. To display an FQDN (Fully Qualified Hostname): **hostname -f**
A fully qualified hostname consists of a short hostname and its DNS domain name. To see only the short name, use the **hostname -s** command. In our case, the FQDN and hostname are the same.

2. Command to check the Linux version: **cat /etc/redhat-release**



1. Basic Directory Commands

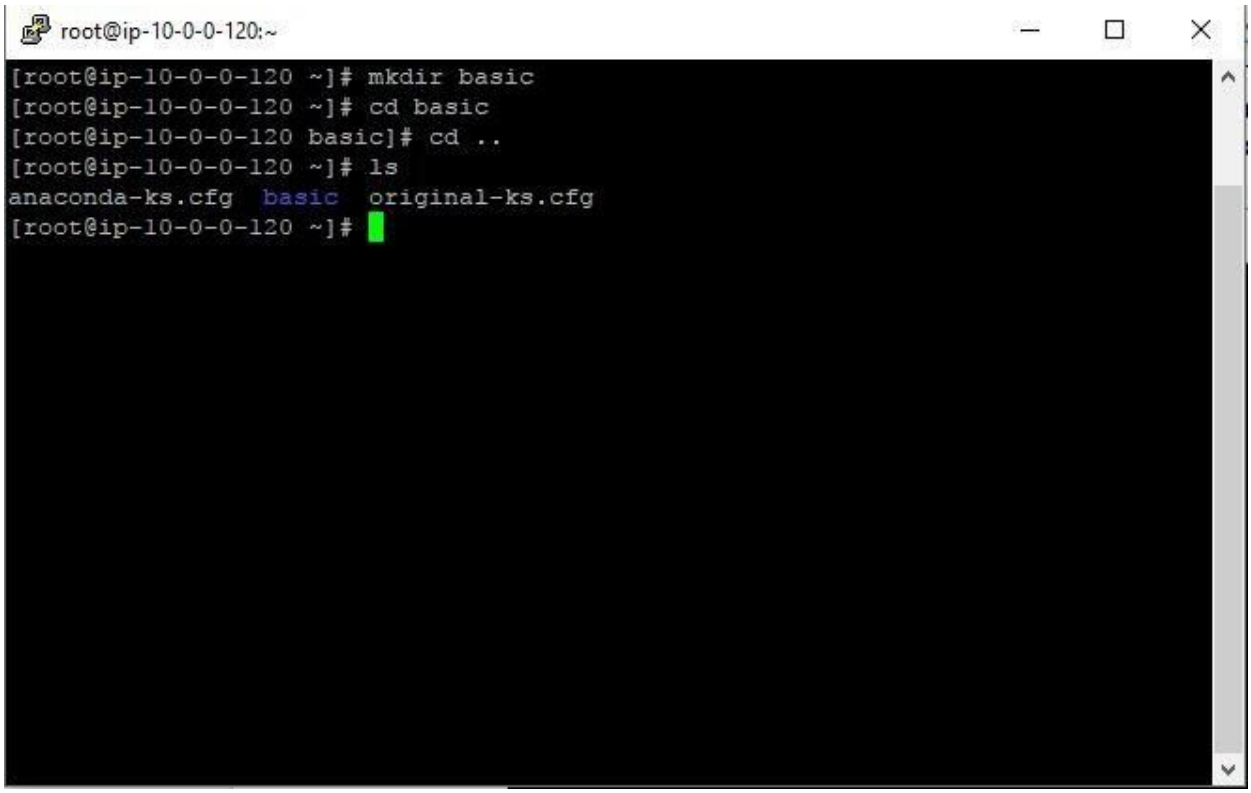
a. **mkdir**: You can use this command to create a new directory within the current directory. The general syntax is 'mkdir <directory name>'.

b. **cd**: 'cd' stands for 'change directory' and you can use this command to change the current working directory. The general syntax is '**cd <directory name or path>'.**

i. **cd ..** : Use this command to go back one directory from the working directory.

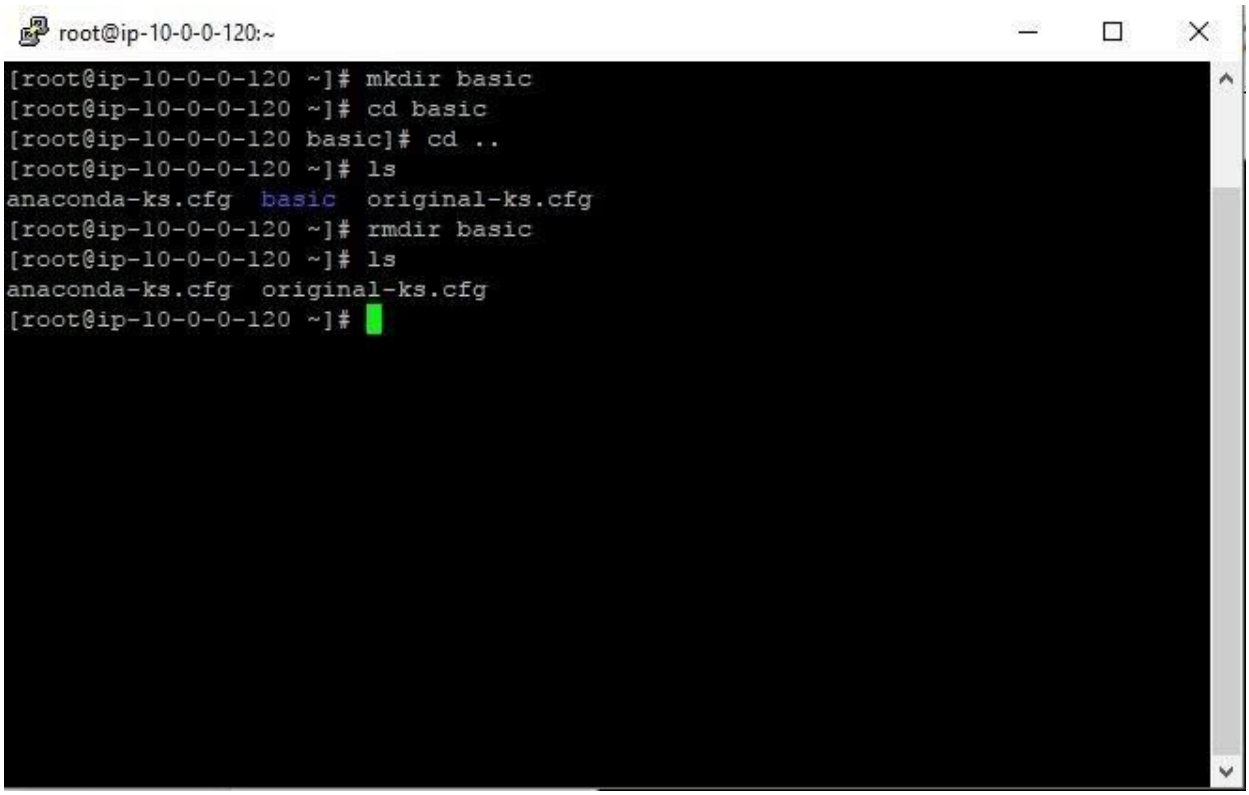
ii. **cd ~** : This command takes you directly to your home directory.

As seen in the example below, we have first created a directory named 'basic' using the **mkdir** command. Then we have changed our current working directory to basic using the **cd** command. Then we used the **cd ..** command to go back one directory. We can check whether our directory was created or not using the **ls** command, which is explained later.

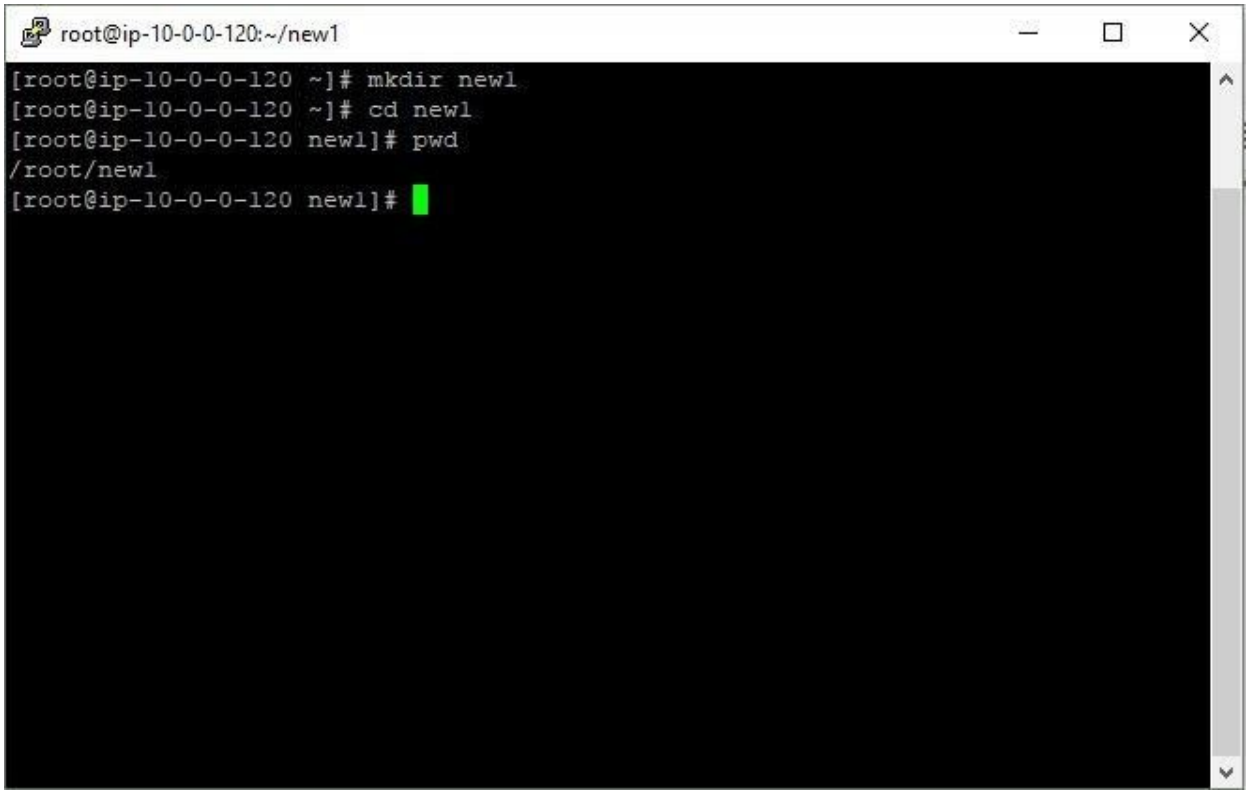


c. **rmdir**: Use this command to remove a directory. The general syntax to remove a directory without files is '**rmdir <directory name>'**. However, if your directory contains files, then the following command is to be used: **rmdir -rv <directory name>**.

To remove our newly created directory, we use the 'rmdir' basic command, as shown in the example below. To verify this, we again use the 'ls' command.



d. **pwd**: It stands for 'print working directory', and it displays the full pathname of the current working directory as its output.
In the example below, you can see that first we created a new directory **new1** using the 'mkdir' command and then used the 'cd' command. Now when you type **pwd** on the terminal, see that it shows the complete path of the current working directory as its output.



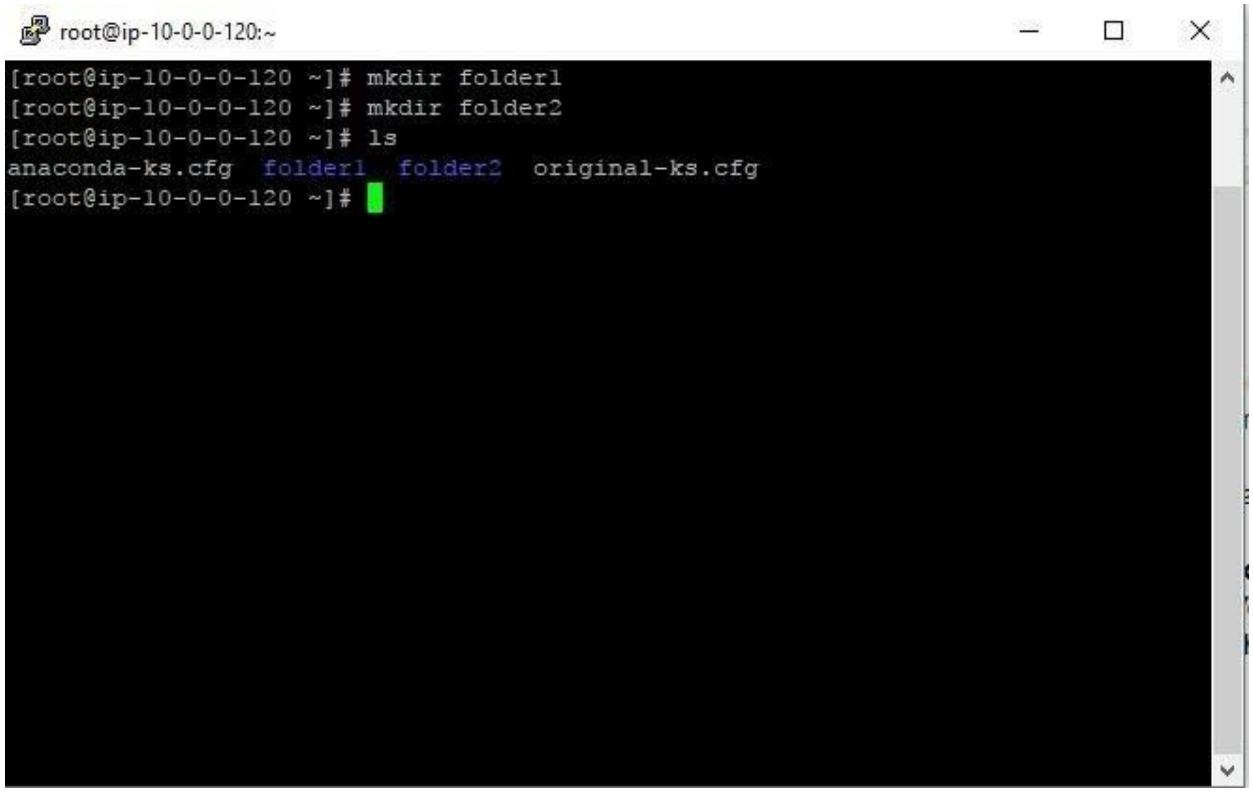
e. **ls**: Use the **ls** command to list all the files and subdirectories in a directory. In Linux, hidden files start with a dot (**.**) and are not visible using the 'ls' command. The '**ls -l**' command is used to view the long output format of the file, which tells you about the permissions, owner, date, etc. You can use the '**ls -a**' command to list all hidden files and subdirectories.

2. **cp**: Use this command to make copies of files and directories.
   a. To copy the content of one file to another, use '**cp <file1> <file2>**'.
   This will copy the contents of 'file1' to 'file2'. If file2 does not exist, the system will create a new file with the name 'file2'.
   b. To copy a file to a directory, use '**cp <filename> <directory name>**'.

   Note that both the file and the directory should exist within the current working directory. If the desired file or directory does not exist in the current directory, you should specify the entire path of the directory.

3. **mv**: Use this command to move files and directories from one place to another.
    a. To move a file from one place to another, use '**mv <filename> <directory name>**'.
    Note that the directory should be present in the current working directory for the command to work. If the directory does not exist in the current working directory, you should specify the entire path of the directory.
    b. The 'mv' command can also be used to rename a directory. This can be done by using '**mv <directory name> <new directory name>**'.

    Task: Before you proceed to the next section, remove all the directories you have created. Then, create two new directories named 'folder1' and 'folder2'. Verify whether the previous directories have been deleted and the new directories created using the 'ls' command.

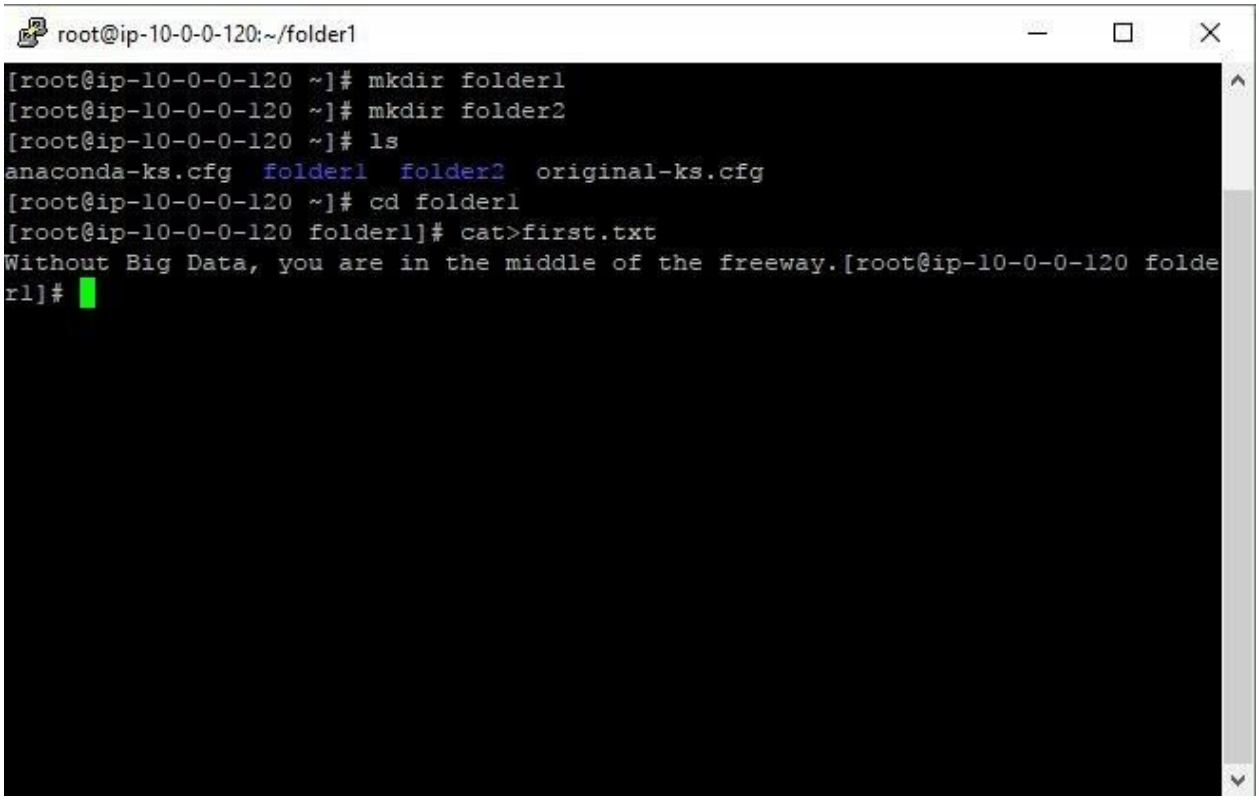


4. Basic file commands

a. Creating a file: There are various ways in which a file can be created on Linux

i. The 'cat' command: To create a file using 'cat', type '**cat>filename**' on the terminal and then enter the text you want in the file. Then, exit using **ctrl+d.**

ii. Use the 'cd' command to navigate to the **folder1** directory. Now, use the **cat** command to create a file named **first.txt**, as shown in the example below. Enter the following text in the file: **Without Big Data, you are blind and deaf in the middle of the freeway**. Now, exit by pressing **ctrl+d twice**. Your file with the required text has now been created.



iii. Using the 'vi' editor: This is the most important command on Linux, which is used to create and edit files.

1. To create a file using the vi editor, type '**vi <filename>**' on the console. If the file with the same name already exists, it will open the current file. Otherwise, it will create a new file.
2. An important aspect of the vi editor is that there are two modes of operation: **command mode** and **insert mode**. When we enter a file using vi, it by default is in the command mode. To enter the

insert mode, we use **i**. The insert mode is used to enter text in the file.

3. To go back to the command mode from the insert mode, press the **escape key**. To exit from a file, type '**:q**' in the command mode. This option will ask you if you want to proceed without saving the changes. If you don't want this option to pop up and directly close the file, then use the **:q!** command. To save a file, use '**:w**'; and to save and exit a file, use the '**:wq**' command.
4. Once you are in the insert mode by pressing i, you can add and delete text from your file. However, you cannot use the arrow keys to navigate through the file. The following keys are used to navigate:
    a. **h** - Move the cursor one character position to the left
    b. **j** - Move the cursor down by a line
    c. **k** - Move the cursor up by a line
    d. **l** - Move the cursor one character position to the right

5. When we press **i**, it will enter text before the cursor position until escape is hit. Instead, if we press **I**, it will enter text at the start of the current line until escape is hit. Similarly, **a** is used to enter text after the current cursor position and **A** to enter text at the end of the current line.
6. **x** is used to delete the character under the current cursor position.
7. **yy** is used to copy the current line into the buffer and **p** is used to paste the contents in the buffer to the line after the current cursor position. To copy n lines, use **nyy** and then use **p** to paste it.
8. There is also an option to search for text in the vi editor. It can be done using **/** and **?**. The **/string** is used to search for the string forward in the file whereas the **?string** is used to search backwards. The **n** and **N** are used to move to the next occurrence of the string in the same and previous direction of the search respectively.

9. Do the following to get familiar with copying and moving files in Linux.
    a. First, open the file first.txt using the vi editor. The command for the same **vi first.txt**. Try editing and navigating through the file using the keys mentioned above. Also, try cutting and pasting lines to get a clear understanding of how vi editor works.

b. Go back to the home directory (home directory is root in our case) using **cd ~** and then use cd to navigate to folder2. Make a file named second.txt using the vi command and add the following text: **Data is the new oil.** Do not forget that to add text we first need to go into the insert mode by pressing **i** Then save and exit the file using

:wq.

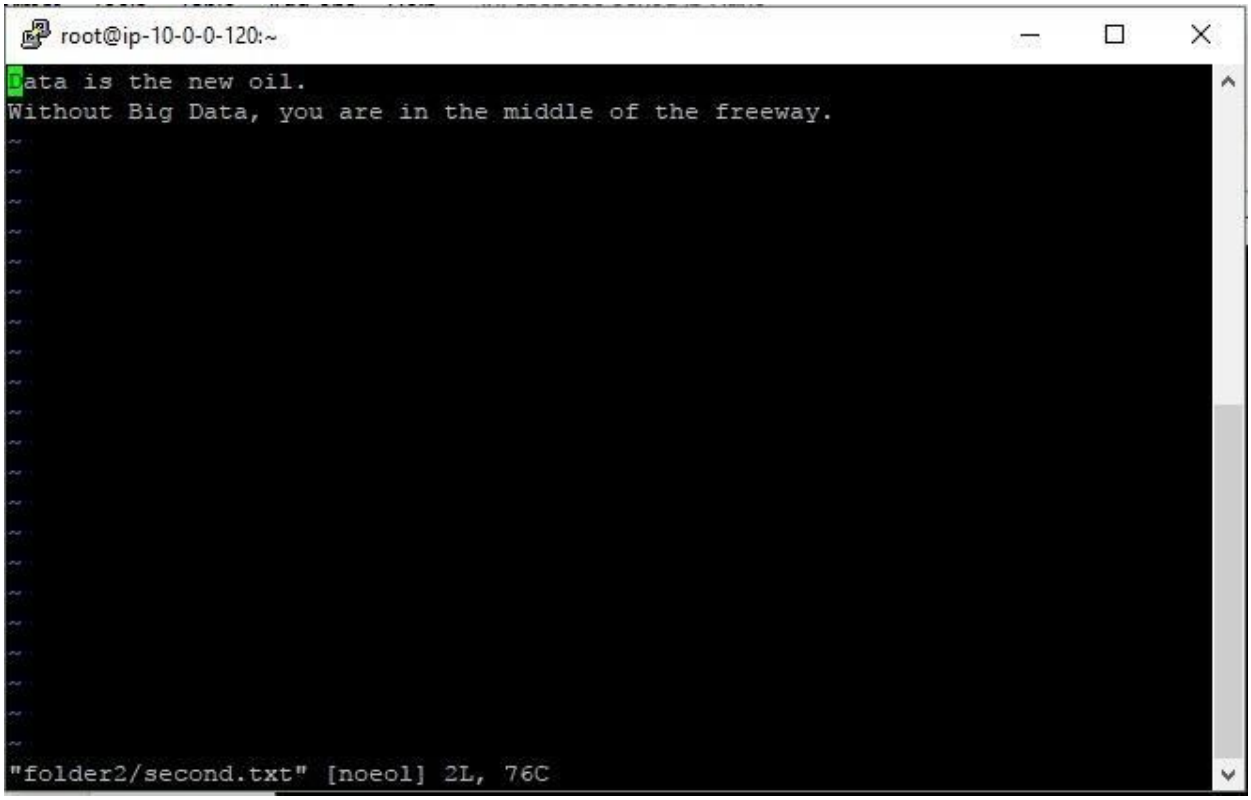
c. Now go to back to the home directory using **cd ~**.

d. Now, to append the contents of first.txt to second.txt, use the following command in the home directory:

cat folder1/first.txt >> folder2/second.txt

You can see that the entire path is specified because the files are inside the sub directories folder1 and folder2.

e. Now go and see the contents of second.txt. You will see that the contents have been appended.

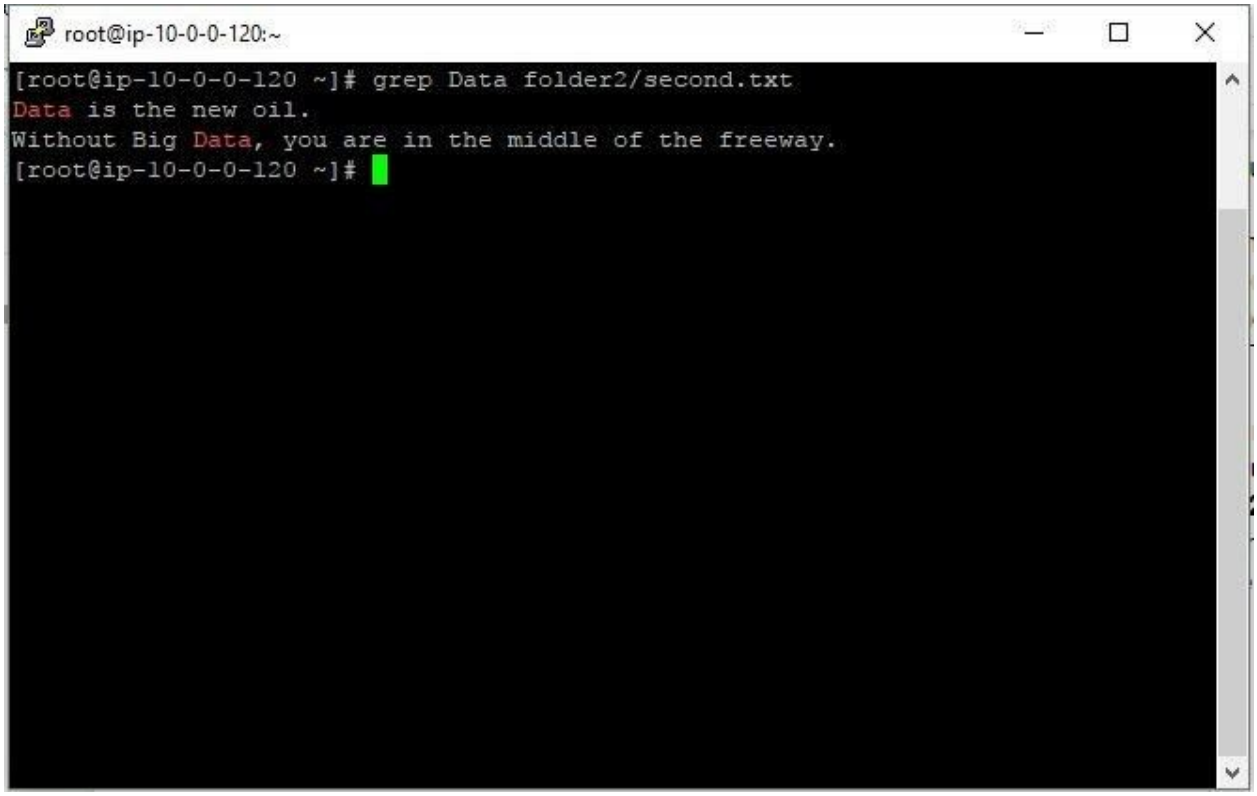


f. The following example is just to make it easier for you to understand the difference between appending and copying. Do not use the following command on the files first.txt and second .txt. We will be using the appended second.txt ahead.
cp folder1/first.txt folder2/second.txt . You will be asked whether you want to overwrite the contents of second.txt. If you press **y**, You can see that the contents of second.txt have been overwritten.

Kindly practice the vi editor to get familiar as it is very important in our course.

5. **grep** command - the grep command is used to search for text in a file. The syntax to use the grep command is **grep pattern filename**. The pattern can be any word or a series of words. There are various options to use along with the grep command. The **grep -n** command also tells us the line number on which the pattern was found. The **grep -i** command can be used to search for a pattern without being case-sensitive.

a. To get familiar with the **grep** command, try finding the word **data** in the file **second.txt**. The command is **grep Data folder2/second.txt** if you are in the **home directory** or **grep Data second.txt** if you are in the **folder2 directory**. As we can see if we are not in the directory in which the file exists, the entire pathname is to be written. To avoid errors, make it a habit to write the entire pathname.

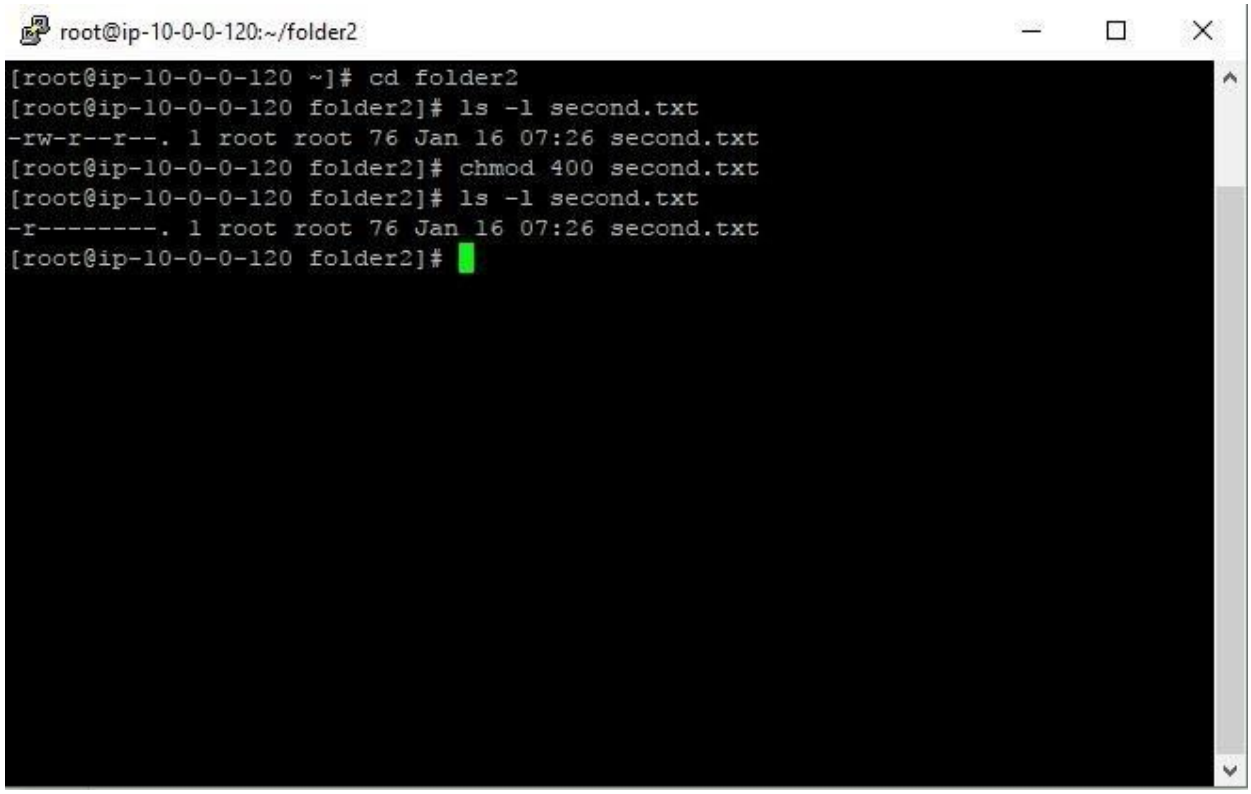


6. **chmod** command - the chmod command is used to change mode or permissions of the file or directory. The syntax for the chmod command is **chmod mode filename**. File control mechanisms are determined by classes and permissions. Each file or directory has three classes: owner, group, other users and three permissions: read(r), write(w) and execute(e).

a. Use cd to move to the folder2 directory. Let us say we used the following command **ls -l second.txt**. The output is shown below.

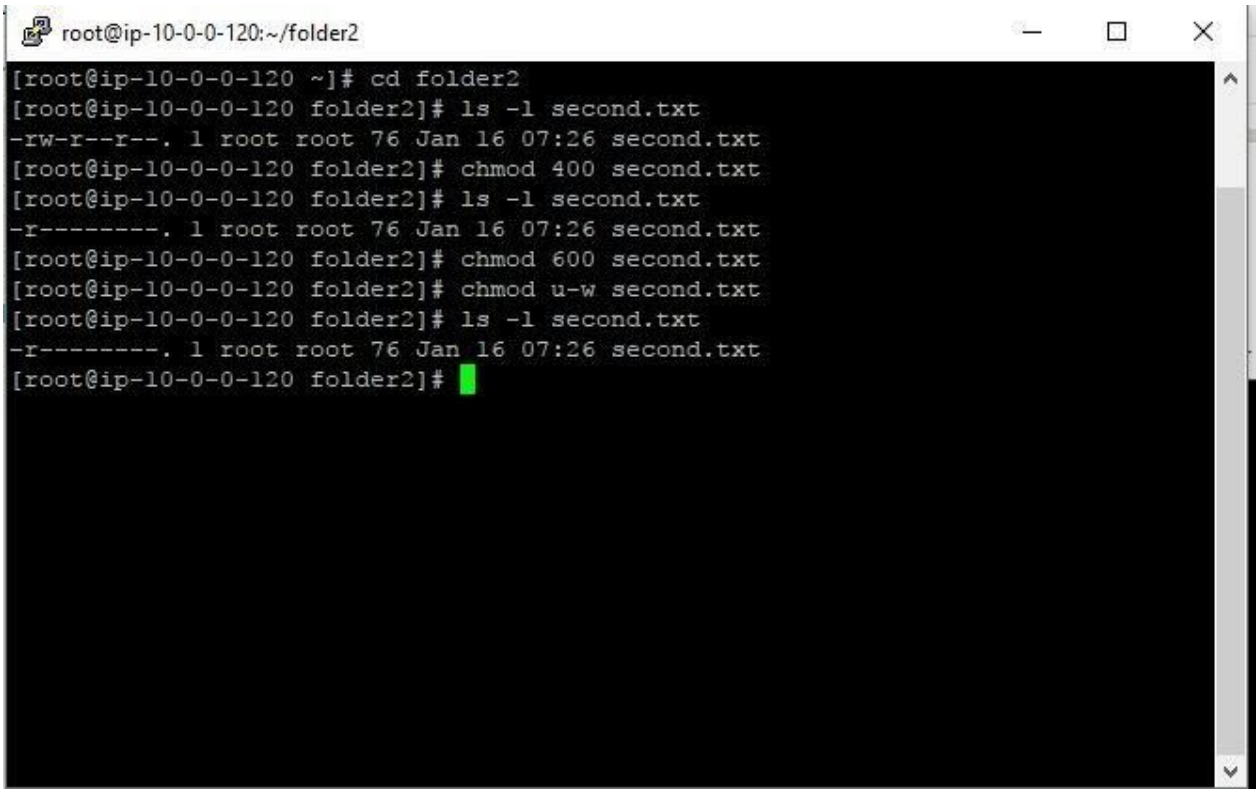
b. Now we need to understand what these 10 bits (**-rw-r--r--**) shown mean. The first bit determines whether it is a file(-) or directory(d). In our case, as we know it is a file(-). The next three bits **rw-** determine the owner permissions, the next three bits **r--** the group permissions and the last three **r--** tell us the permission for other users. In our case, the 10 bits indicate that the owner has read and write permissions, the group and other users have only read permissions.

c. The mode can be changed either in the numeric or symbolic format. In the numeric format, 4 stands for read, 2 for write, and 1 for execute. In the numeric format, the mode always has 3 digits. The first for the owner, the second for the group and the third for others. Each digit is determined by the sum of all the permissions granted. For example, if the user has read, write and execute permissions, the digit corresponding to that will be 4+2+1=7. Now let us execute the following command in the terminal: **chmod 400 second.txt**. Now type the command **ls -l second.txt**. As seen, now the owner has only read permissions while no permission is granted to group and others.



d. To change the mode in the symbolic mode, we need to understand the meaning of a few symbols: + means add permissions, - means remove permissions, = means adding permissions and removing permissions of unspecified fields, u - user(owner), g - group, o- others and a - all. For example, chmod u-x means denying execute permissions to the user.

e. As you can see below, I have first given the owner read and write permissions using **chmod 600 second.txt** and then used **chmod u-w** to deny the write permissions.

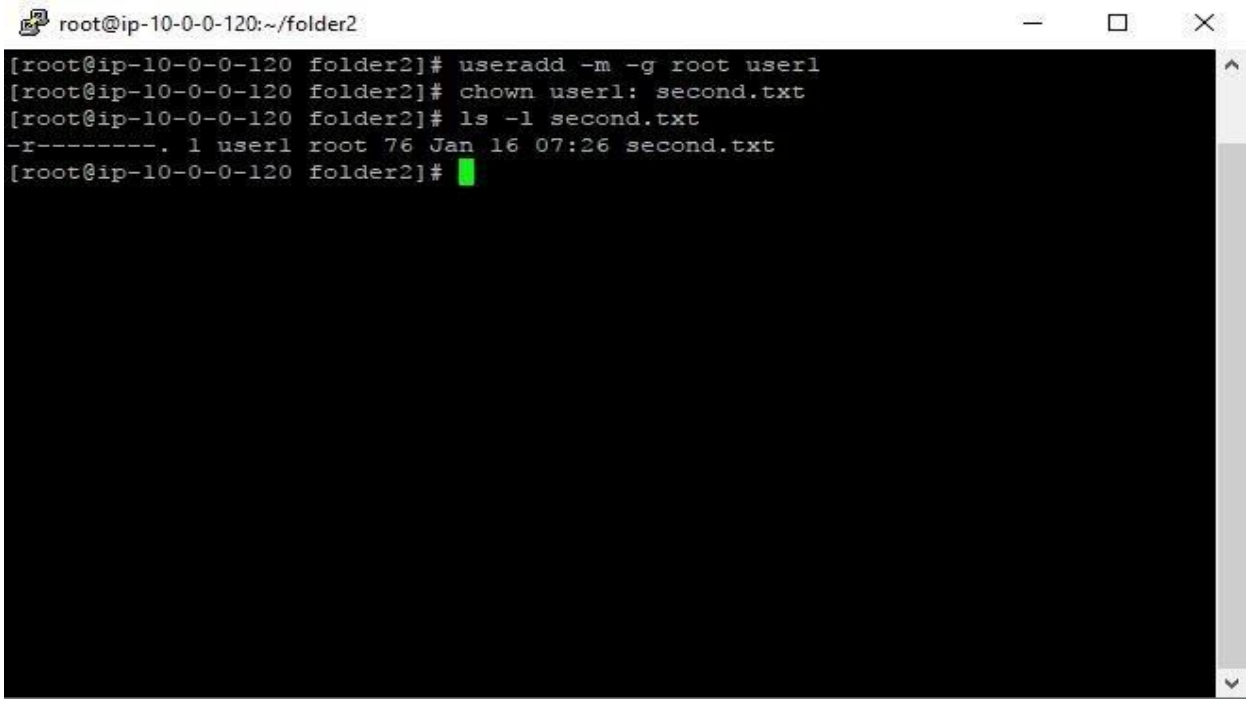


7. chown command - this command is used to change the user or group ownership of the given file. The general syntax for the chown command is **chown [OWNER][:[GROUP] filename**.
   a. Go through this link to understand what a user and a group are, and learn how to add a user to a group. Let us take an example of the second.txt file.
   b. Add a user named user1 to the root group. The command is

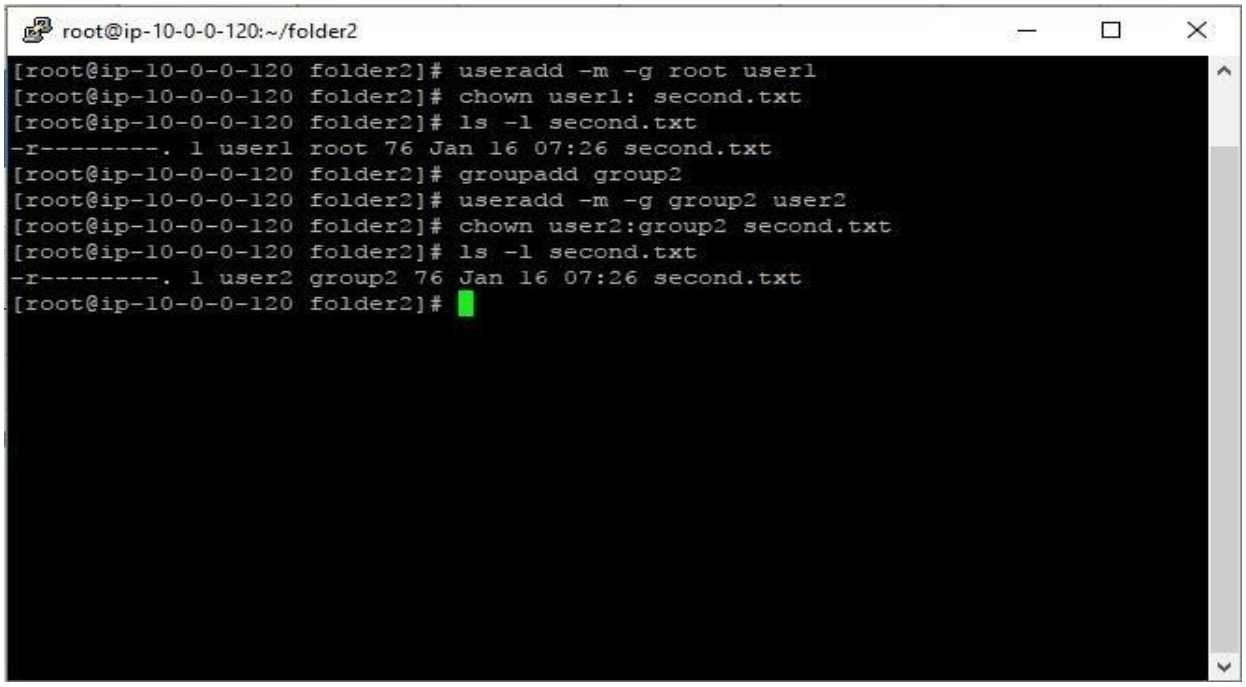
useradd -m -g root user1

   c. Now, use the **ls -l second.txt** to see the owner and the group of the file. As we can see in the image above, the owner, as well as the group of the file, are both root.
   d. Now let us use the chown command to change the user of the file without changing its group. The command for the same is **chown user1: second.txt.**

e. Use the **ls-l** command to verify whether the user has changed or not.

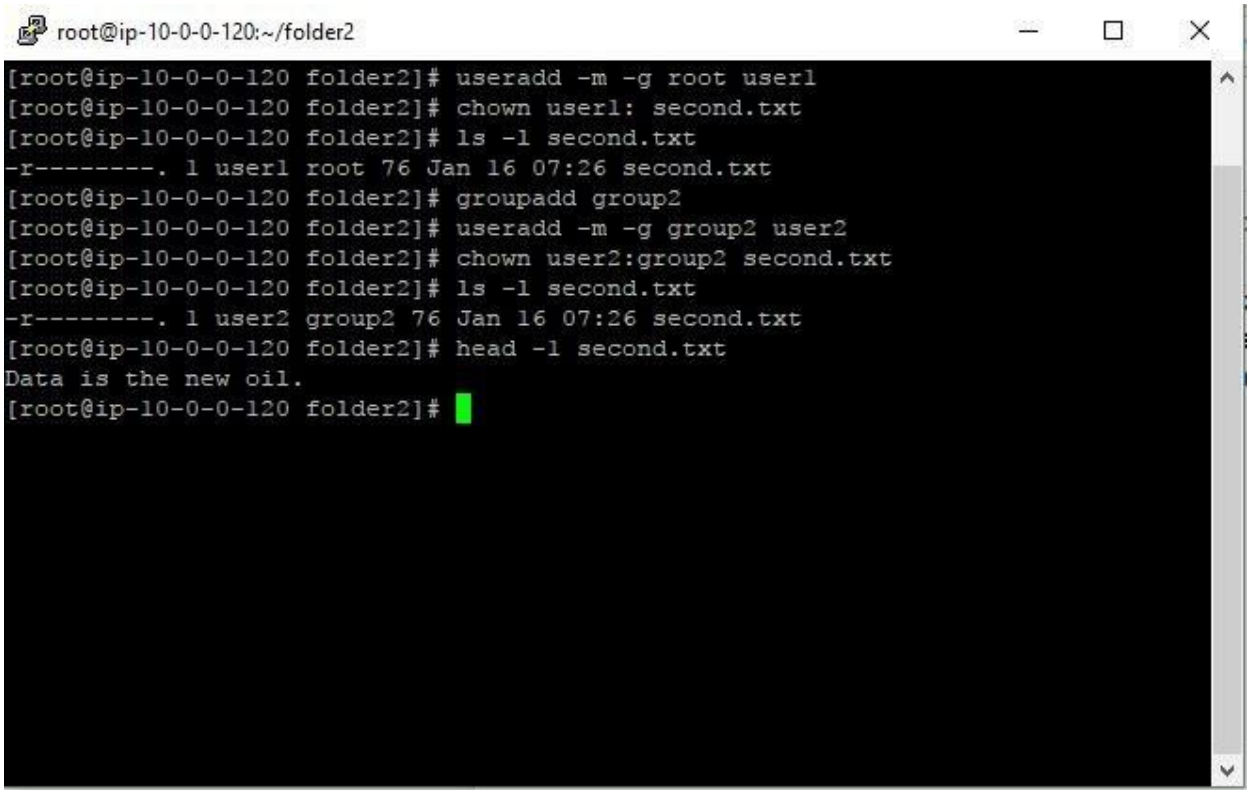


f. To change the group, you need to first create a new group. Let us create a group named **group2**. Add a user named **user2** to **group2** and change the owner as well as the group of the file **second.txt**. The command to create a new group is **groupadd group2.**



Practice the commands in the link above .

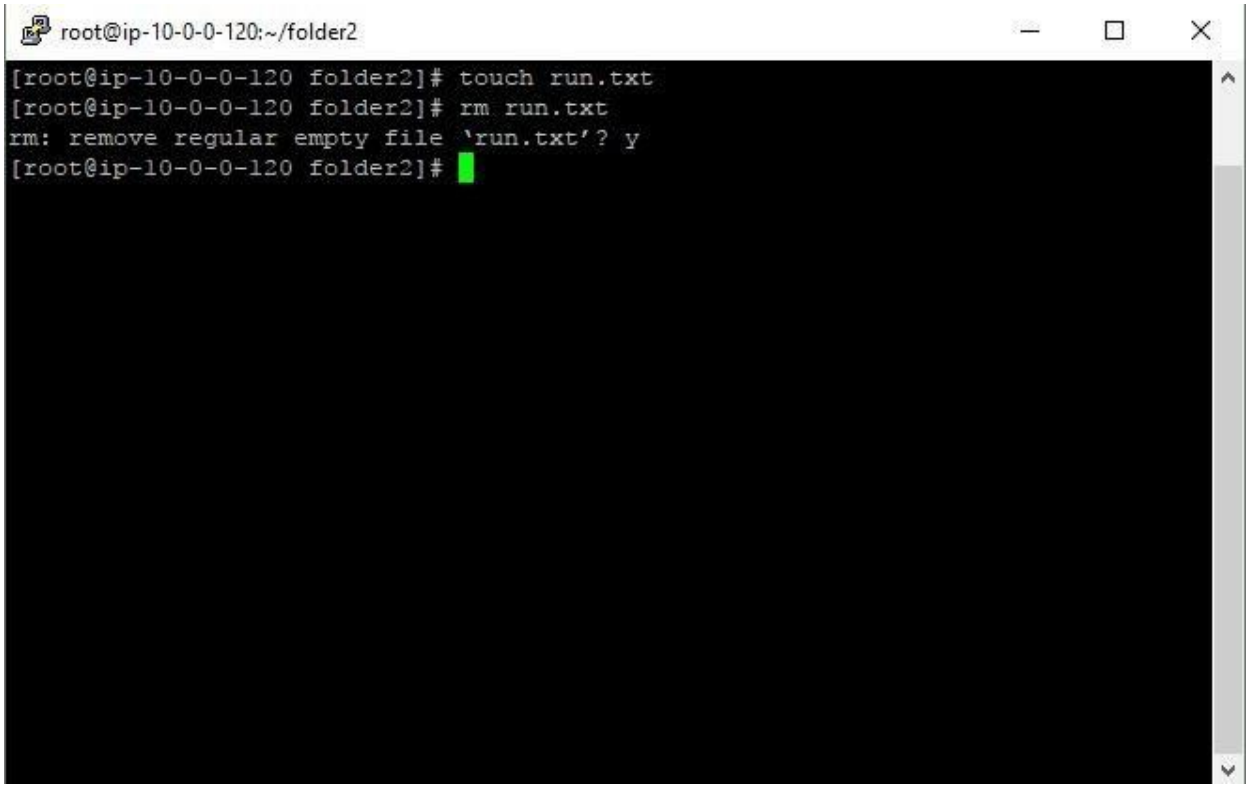
8. Head command - the **head -n <filename/filenames>** command displays the specified number of lines from the file to the output. For example, **head -1 second.txt** displays the first line of the file to the output.



9. tail command: the **tail -n <filename>** command is used to the specified number of lines from the bottom of the file to the output. For example, **tail -1 second.txt** will display the last line of the file

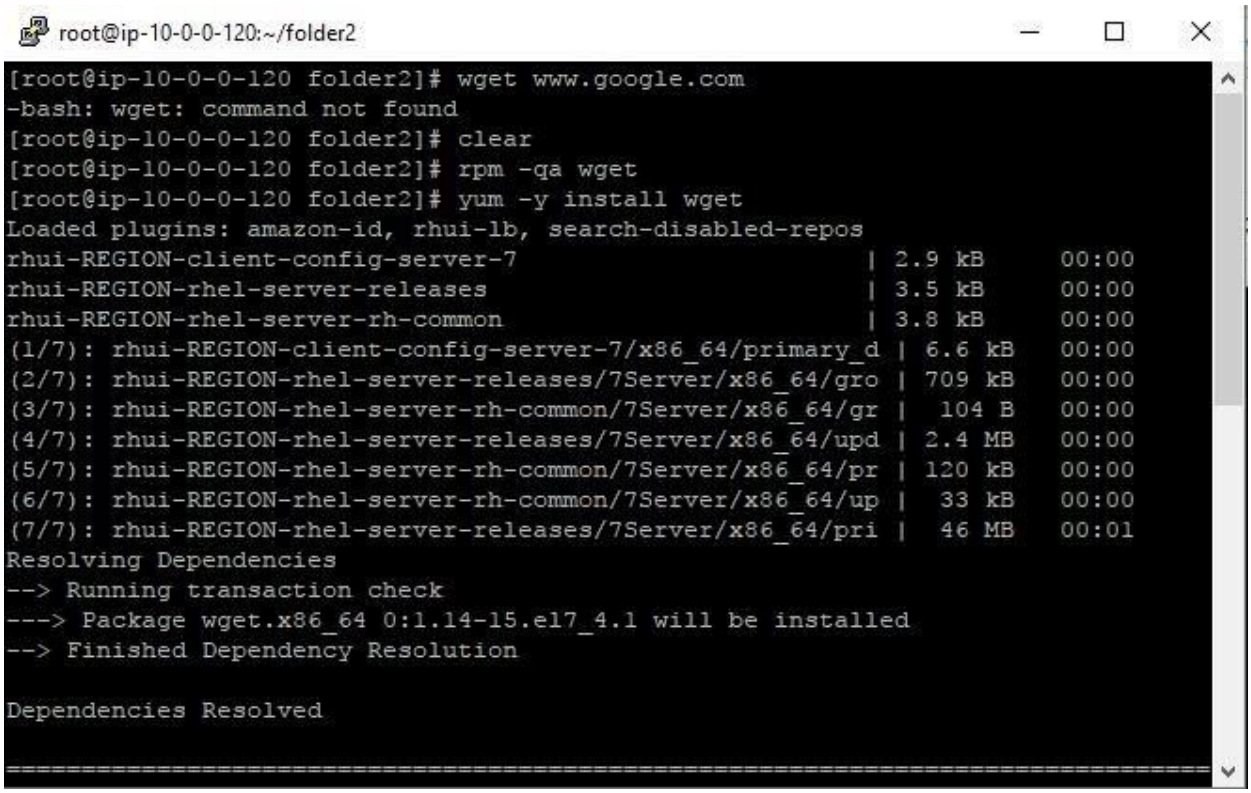
Please note that if the number of lines is not specified in the command for the head and tail command, it by default will display the first and last 10 lines.

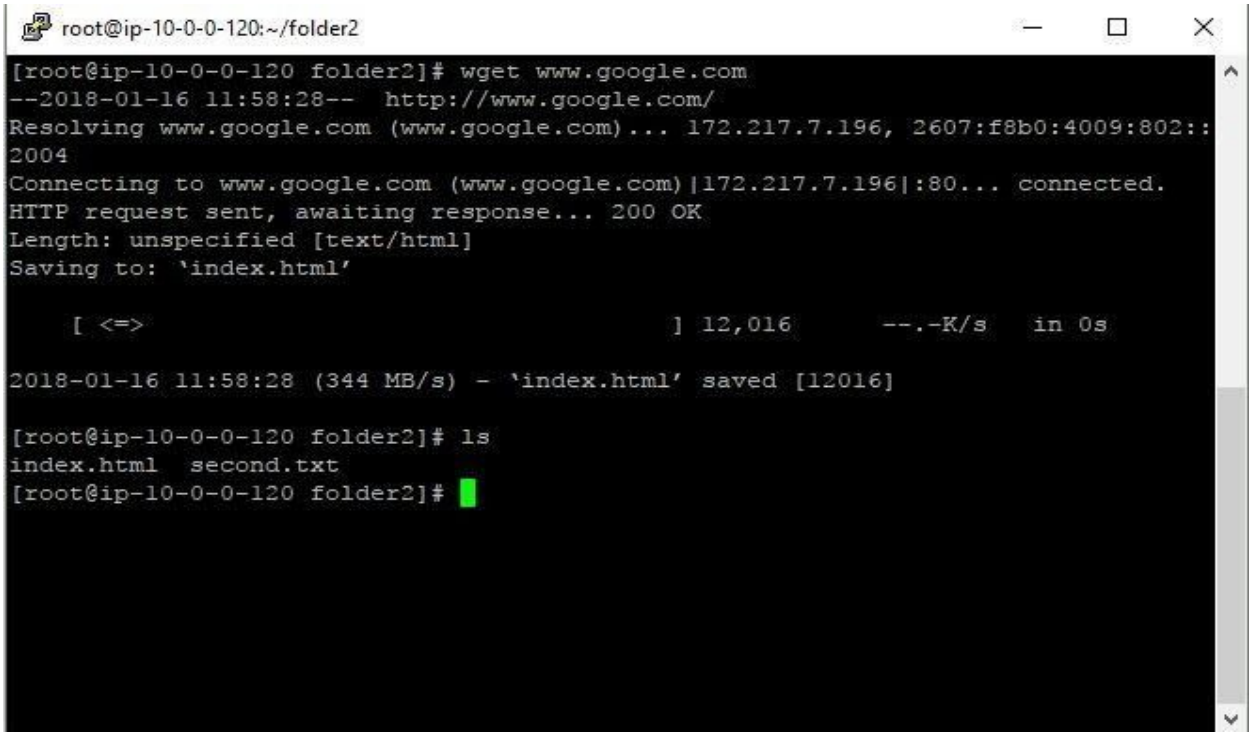
10. **rm** command - the **rm <filename>** command is used to delete files. For example, create a file with any random name and use the rm command to delete it. I have created a file named **run.tx**t and used the **rm** command to delete it. We can use cat to do so or use the **touch <filename>** command. The touch command is used to create



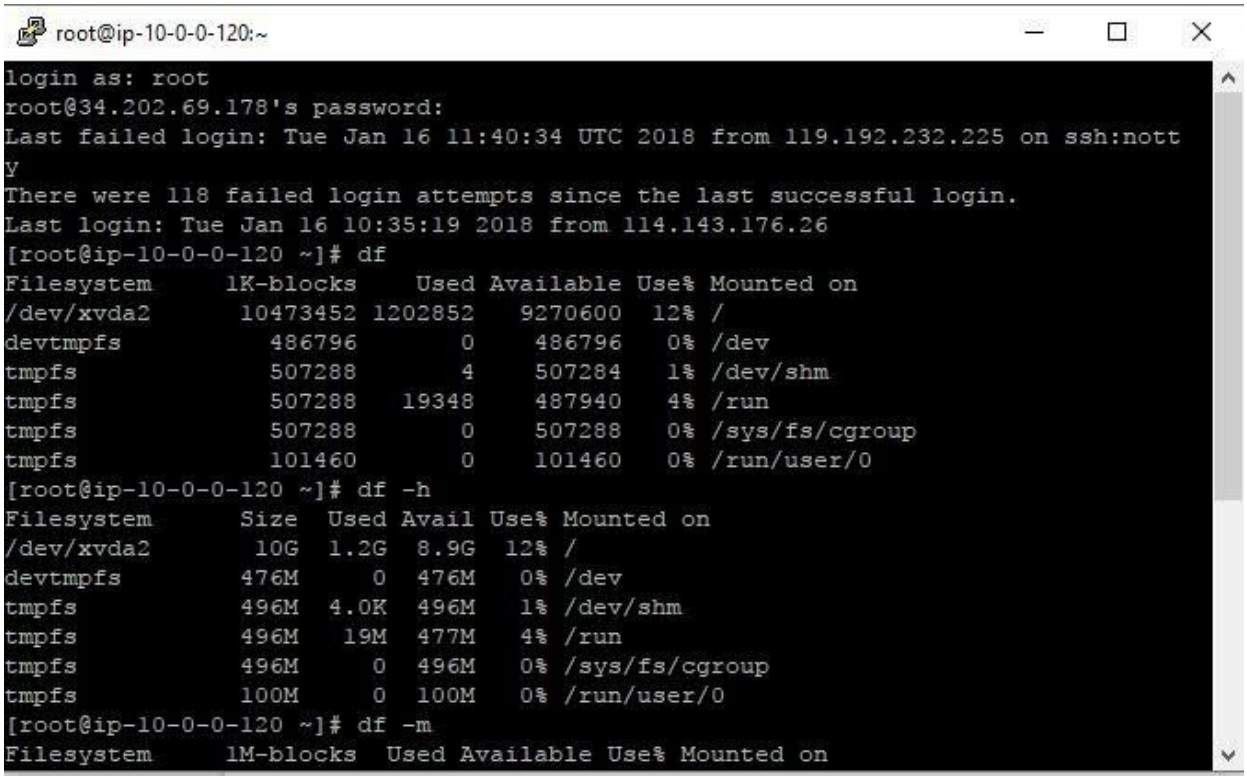
empty files.

11. **wget** command - the **wget** command is used to download files from the internet. Kindly use **wget <link>** to download a file or page from the internet. To run the wget command in the background, use **wget -b <link>**. For example, I have downloaded the www.google.com web page using the wget command. Before using the wget command, we need to check whether the **wge**t utility is installed or not. This can be checked using the **rpm -qa wget** command. If installed, it will tell the wget utility version number. Otherwise, it will do nothing. To install wget, use **yum -y install wget**. The yum command is used to update, download, and search for any software. The basic syntax for the yum command is

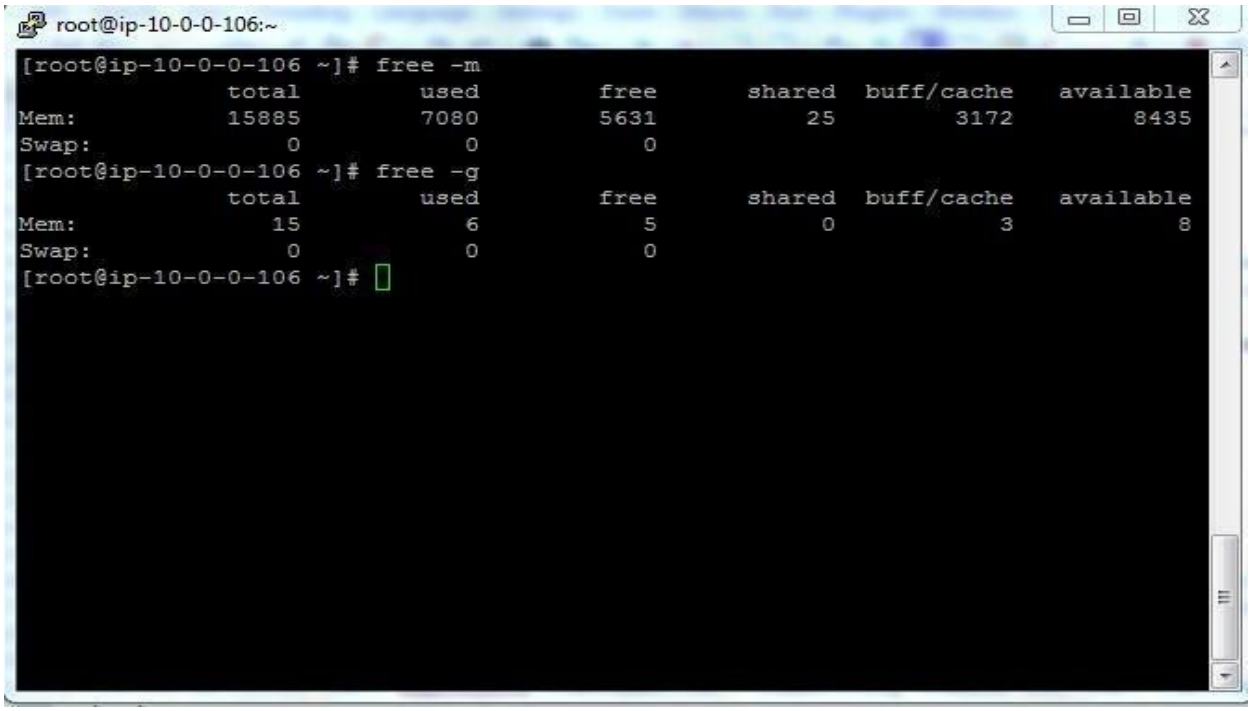




12. Disk Usage Commands in Linux
    a. df - the df command is used to check the amount of disk space used and available in the Linux file system.
        i. df - h : the -h converts the df table in human readable format.



13. Free command - the free command is used to check the amount of free space present in memory. To display in in MB, use free -m as shown below. We can also display the memory in GB using free -g.



We can also clear the cache using the echo 3 > /proc/sys/vm/drop_caches command as shown in the image below.