Q1: Revise file permission & user management

Answer: **Owner permissions:** The owner's permissions determine what actions the owner of the file can perform on the file.

**Group permissions:** The group's permissions determine what actions a user, who is a member of the group that a file belongs to, can perform on the file.

**Other (world) permissions** The permissions for others indicate what action all other users can perform on the file.

While using ls -l command, it displays various information related to file permission as follows:

*ls -l /home/amrood*

*-rwxr-xr-- 1 amrood users 1024 Nov 2 00:10 myfile*

*drwxr-xr--- 1 amrood users 1024 Nov 2 00:10 mydir*

the first column represents different access modes, i.e., the permission associated with a file or a directory.

The permissions are broken into groups of threes, and each position in the group denotes a specific permission, in this order: read (r), write (w), execute (x)

The first three characters (2-4) represent the permissions for the file's owner. For example, -rwxr-xr-- represents that the owner has read (r), write (w) and execute (x) permission.

The second group of three characters (5-7) consists of the permissions for the group to which the file belongs. For example, -rwxr-xr-- represents that the group has read (r) and execute (x) permission, but no write permission.

The last group of three characters (8-10) represents the permissions for everyone else. For example, -rwxr-xr-- represents that there is read (r) only permission.

**File Access Modes**

The permissions of a file are the first line of defense in the security of a Unix system. The basic building blocks of Unix permissions are the read, write, and execute permissions, which have been described below

**Read**

Grants the capability to read, i.e., view the contents of the file.

**Write**

Grants the capability to modify, or remove the content of the file.

**Execute**

User with execute permissions can run a file as a program.

**Directory Access Modes**

Directory access modes are listed and organized in the same manner as any other file. There are a few differences that need to be mentioned âˆ’

**Read**

Access to a directory means that the user can read the contents. The user can look at the filenames inside the directory.

**Write**

Access means that the user can add or delete files from the directory.

**Execute**

Executing a directory doesn't really make sense, so think of this as a traverse permission.A user must have execute access to the bin directory in order to execute the ls or the cd command.

Q2 create a user in linux with userid as 10001

**Answer** --> *sudo useradd -u 10001 laser*

- Add the user into the secondary group {sales} at time of creating the user

**Answer** --> sudo useradd -G 10001 sales

Q3: Try to move the file from local environment to any folder into hadoop ecosystem.

**Answer** --> hdfs dfs -put 4march.txt /deep or

*hdfs dfs -copyFromLocal 4march.txt /deep*

**To verify**: hdfs dfs -ls /deep

Q4: create a empty file in any locaiton in hadoop

**Answer** --> hdfs dfs -touchz /deep/7-march.txt

Q5 Change the permission of the folder & the files & subdirectories present inside the location of hadoop

- After that also verify the folder & also the internal files inside the folder

**Answer** --> hdfs dfs -chmod 654 /deep/test

Q6 change the replicaiton factor for any file present in hadoop to 3

- Also change the replication factor for any file while using the {put command } in hadoop to 3

**Answer** --> hdfs dfs -setrep -R 3 /filename

hdfs dfs -Ddfs.replication=3 -put /home/cloudera/Desktop/

Q7 - Create any folder in hadoop

- Move the file from local to hadoop folder

**Answer** --> hdfs dfs -copyFromLocal 4march.txt /deep

- create one more file with data into local linux

**Answer** -->

- Try to copy the contents of new file from local machien to the hadoop environment now in the same folder

**Answer** -->