HDFS Commands - GeeksforGeeks

HDFS - javatpoint

HDFS Commands

HDFS is the primary or major component of the Hadoop ecosystem which is responsible for storing large data sets of structured or unstructured data across various nodes and thereby maintaining the metadata in the form of log files. To use the HDFS commands, first you need to start the Hadoop services using the following command:

```
sbin/start-all.sh
```

To check the Hadoop services are up and running use the following command:

```
jps
```

Commands:

1. **ls:** This command is used to list all the files. Use *lsr* for recursive approach. It is useful when we want a hierarchy of a folder.

   Syntax:

   ```
   hadoop  fs  -ls  <path>
   ```

   Example:

   ```
   bin/hdfs dfs -ls /
   ```

   It will print all the directories present in HDFS. bin directory contains executables so, *bin/hdfs* means we want the executables of hdfs particularly *dfs*(Distributed File System) commands.

2. **mkdir**: To create a directory. In Hadoop *dfs* there is no home directory by default. So let's first create it.

   Syntax:

   ```
   bin/hdfs dfs -mkdir <folder name>


   creating home directory:


   hdfs/bin -mkdir /user

   hdfs/bin -mkdir /user/username -> write the username of your computer
   ```

   Example:

   ```
   bin/hdfs dfs -mkdir  /geeks  =>  '/' means absolute path

   bin/hdfs dfs -mkdir  geeks2  =>   Relative path -> the folder will be

                                     created relative to the home directory.
   ```

3. **touchz**: It creates an empty file.

Syntax:

```
bin/hdfs dfs  -touchz  <file_path>
```

Example:

```
bin/hdfs dfs -touchz  /geeks/myfile.txt
```

4. **copyFromLocal (or) put:** To copy files/folders from local file system to hdfs store. This is the most important command. Local filesystem means the files present on the OS.

Syntax:

```
bin/hdfs dfs -copyFromLocal <local file path>  <dest(present on hdfs)>
```

Example: Let's suppose we have a file *AI.txt* on Desktop which we want to copy to folder *geeks* present on hdfs.

```
bin/hdfs dfs -copyFromLocal ../Desktop/AI.txt /geeks


(OR)


bin/hdfs dfs -put ../Desktop/AI.txt /geeks
```

5. **cat:** To print file contents.

Syntax:

```
bin/hdfs dfs -cat <path>
```

Example:

```
// print the content of AI.txt present

// inside geeks folder.

bin/hdfs dfs -cat /geeks/AI.txt ->
```

6. **copyToLocal (or) get:** To copy files/folders from hdfs store to local file system.

Syntax:

```
bin/hdfs dfs -copyToLocal  <<srcfile(on hdfs)> <local file dest>
```

Example:

```
bin/hdfs dfs -copyToLocal  /geeks   ../Desktop/hero


(OR)


bin/hdfs dfs -get /geeks/myfile.txt  ../Desktop/hero
```

myfile.txt from *geeks* folder will be copied to folder *hero* present on *Desktop*.

Note: Observe that we don't write *bin/hdfs* while checking the things present on local filesystem.

7. **moveFromLocal:** This command will move file from local to hdfs.

Syntax:

```
bin/hdfs dfs -moveFromLocal <local src>   <dest(on hdfs)>
```

Example:

```
bin/hdfs dfs -moveFromLocal  ../Desktop/cutAndPaste.txt   /geeks
```

8. **cp:** This command is used to copy files within hdfs. Lets copy folder *geeks* to *geeks_copied*.

   Syntax:

   ```
   bin/hdfs dfs -cp  <src(on hdfs)>  <dest(on hdfs)>
   ```

   Example:

   ```
   bin/hdfs -cp /geeks  /geeks_copied
   ```

9. **mv:** This command is used to move files within hdfs. Lets cut-paste a file *myfile.txt* from *geeks* folder to *geeks_copied*.

   Syntax:

   ```
   bin/hdfs dfs -mv  <src(on hdfs)> <src(on hdfs)>
   ```

   Example:

   ```
   bin/hdfs  -mv  /geeks/myfile.txt  /geeks_copied
   ```

10. **rmr:** This command deletes a file from HDFS *recursively*. It is very useful command when you want to delete a *non-empty directory*.

    Syntax:

    ```
    bin/hdfs dfs -rmr <filename/directoryName>
    ```

    Example:

    ```
    bin/hdfs dfs -rmr  /geeks_copied -> It will delete all the content inside the
                                        directory then the directory itself.
    ```

11. **du:** It will give the size of each file in directory.

    Syntax:

    ```
    bin/hdfs dfs -du  <dirName>
    ```

    Example:

    ```
    bin/hdfs dfs -du /geeks
    ```

12. **dus:**: This command will give the total size of directory/file.

    Syntax:

    ```
    bin/hdfs dfs -dus  <dirName>
    ```

    Example:

    ```
    bin/hdfs dfs -dus /geeks
    ```

13. **stat:** It will give the last modified time of directory or path. In short it will give stats of the directory or file.

    Syntax:

```
bin/hdfs  dfs -stat    <hdfs file>
```

Example:

```
bin/hdfs dfs -stat /geeks
```

14. **setrep:** This command is used to change the replication factor of a file/directory in HDFS. By default it is 3 for anything which is stored in HDFS (as set in hdfs *core-site.xml*).

    Example 1: To change the replication factor to 6 for *geeks.txt* stored in HDFS.

    ```
    bin/hdfs dfs -setrep -R -w 6 geeks.txt
    ```

    Example 2: To change the replication factor to 4 for a directory *geeksInput* stored in HDFS.

    ```
    bin/hdfs dfs -setrep -R  4 /geeks
    ```

    Note: The **-w** means wait till the replication is completed. And **-R** means recursively, we use it for directories as they may also contain many files and folders inside them.

Note: There are more commands in HDFS but we discussed the commands which are commonly used when working with Hadoop. You can check out the list of *dfs* commands using the following command:

```
bin/hdfs dfs
```

Hadoop copyFromLocal command

Hadoop **copyFromLocal** command is used to copy the file from your local file system to the HDFS(Hadoop Distributed File System). *copyFromLocal* command has an optional switch *–f* which is used to replace the already existing file in the system, means it can be used to update that file. *-f* switch is similar to first delete a file and then copying it. If the file is already present in the folder then copy it into the same folder will automatically throw an error.
Syntax to copy a file from your local file system to HDFS is given below:

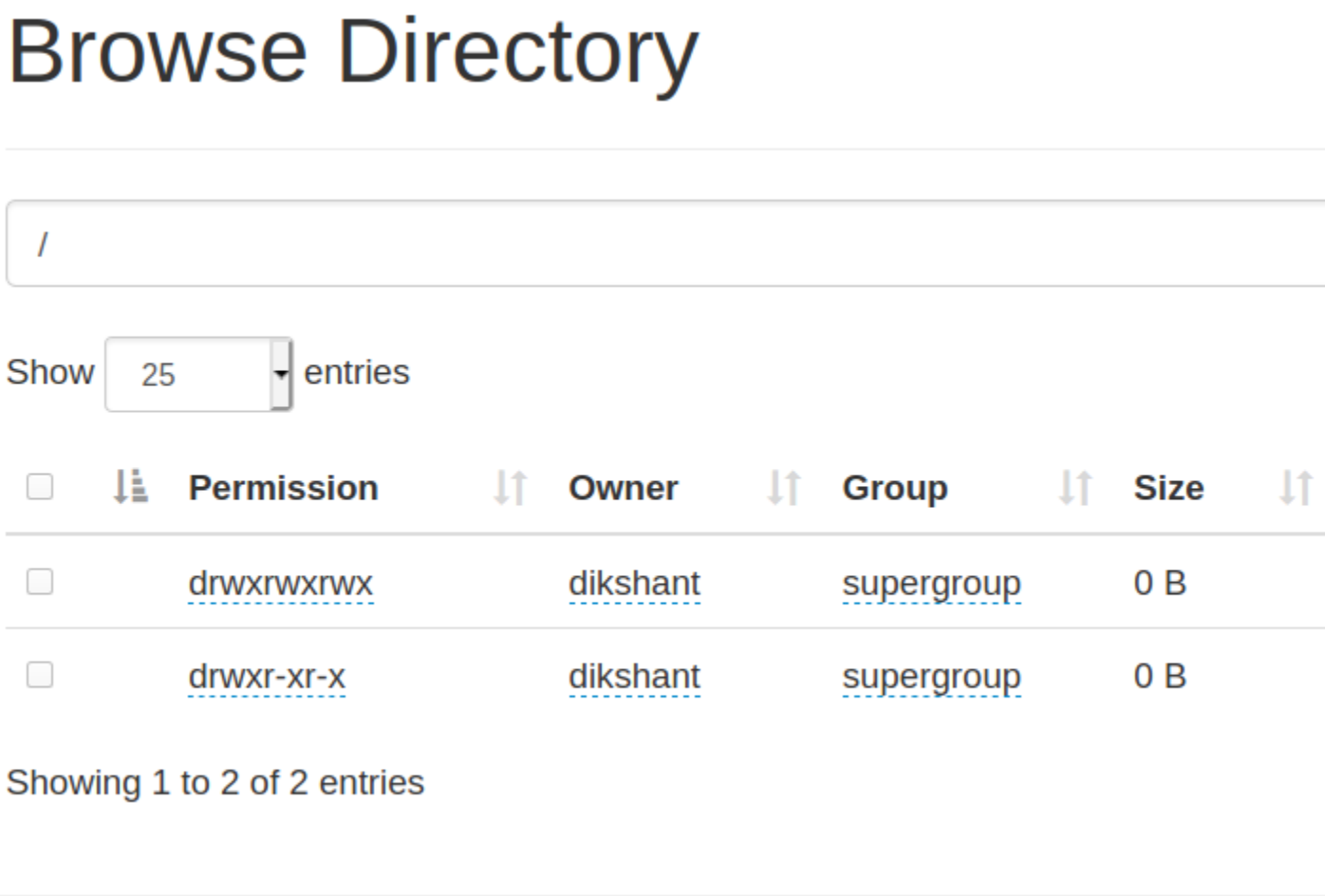
```
hdfs dfs -copyFromLocal /path 1 /path 2 .... /path n /destination
```

The *copyFromLocal* local command is similar to the *-put* command used in HDFS. we can also use *hadoop fs* as a synonym for *hdfs dfs*. The command can take multiple arguments where all the paths provided are of the source from where we want to copy the file except the last one which is the destination, where the file is copied. Make sure that the destination should be a directory.
Our objective is to copy the file from our local file system to HDFS. In my case, I want to copy the file name *Salaries.csv* which is present at */home/dikshant/Documents/hadoop_file* directory.

```
dikshant@dikshant-Inspiron-5567:~$ cd /home/dikshant/Documents/ha
dikshant@dikshant-Inspiron-5567:~/Documents/hadoop_file$ ls
Salaries.csv
dikshant@dikshant-Inspiron-5567:~/Documents/hadoop_file$
```

Steps to execute copyFromLocal Command

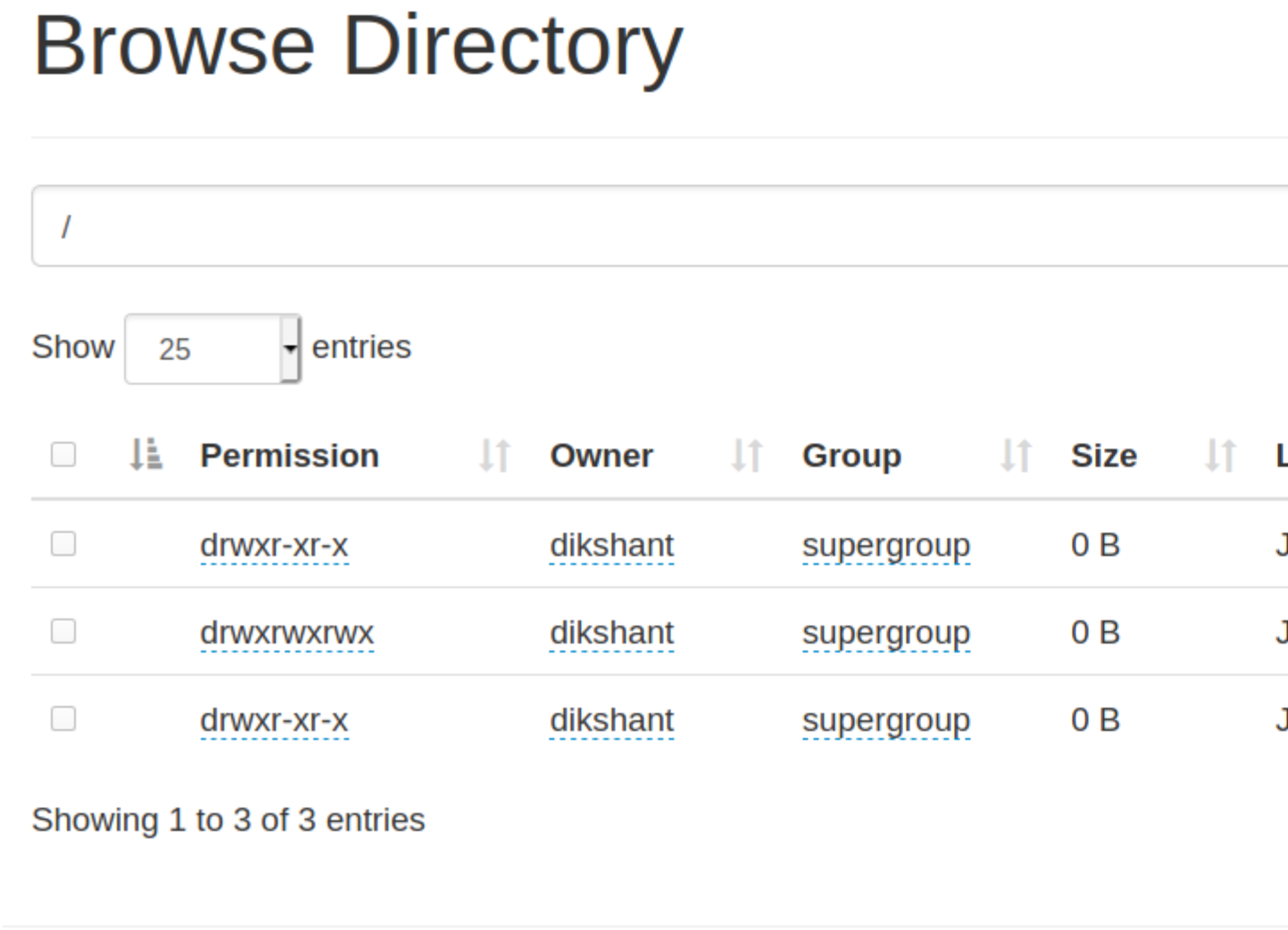
Let's see the current view of my *Root* directory in HDFS.



Step 1: Make a directory in HDFS where you want to copy this file with the below command.

```
hdfs dfs -mkdir /Hadoop_File
```

```
dikshant@dikshant-Inspiron-5567:~$ hdfs dfs -mkdir /Hadoop_File
dikshant@dikshant-Inspiron-5567:~$
```



Step 2: Use copyFromLocal command as shown below to copy it to HDFS */Hadoop_File* directory.

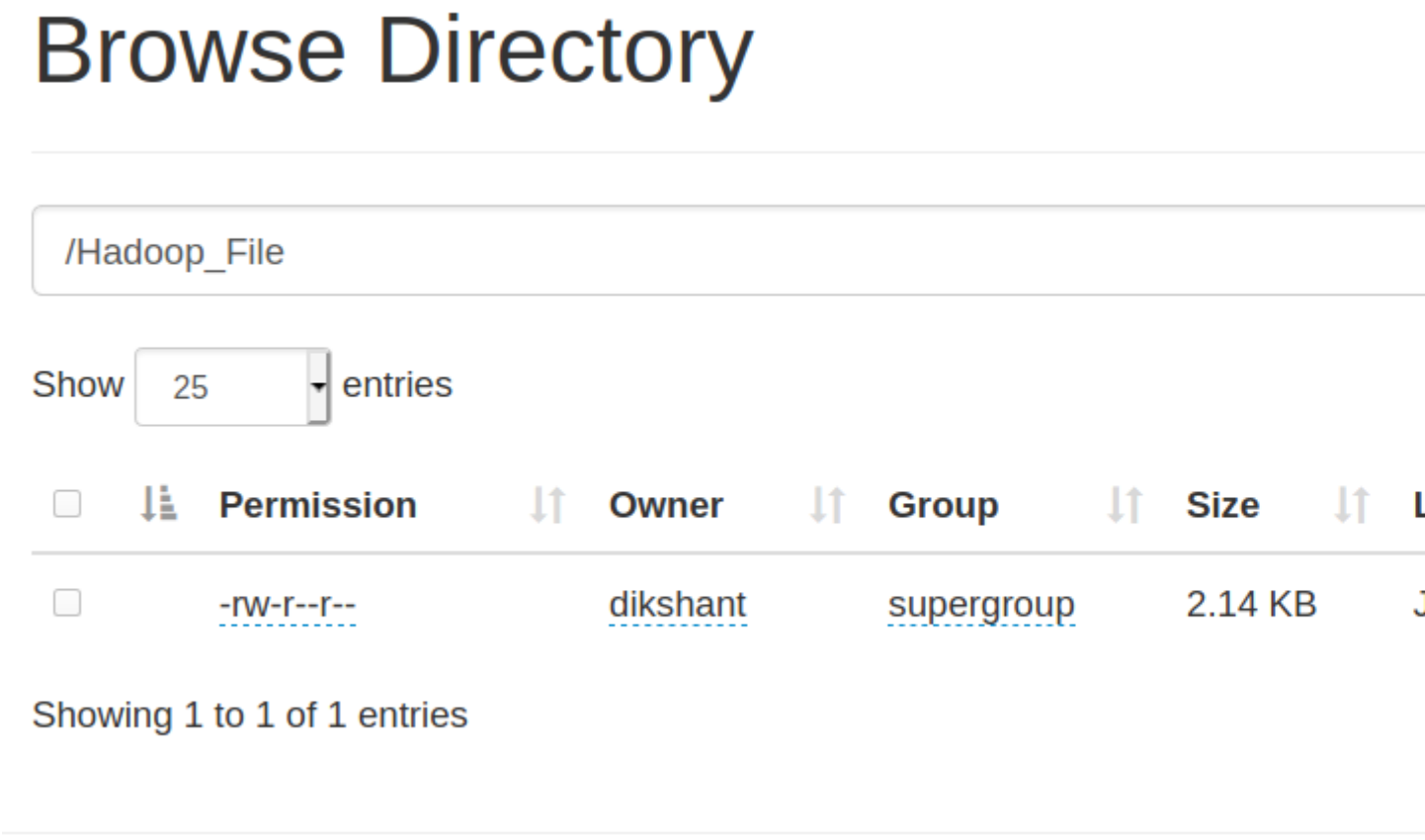
```
hdfs dfs -copyFromLocal
/home/dikshant/Documents/hadoop_file/Salaries.csv /Hadoop_File
```

```
dikshant@dikshant-Inspiron-5567:~$ hdfs dfs -copyFromLocal /home/
p_File
dikshant@dikshant-Inspiron-5567:~$
```

Step 3: Check whether the file is copied successfully or not by moving to its directory location with below command.

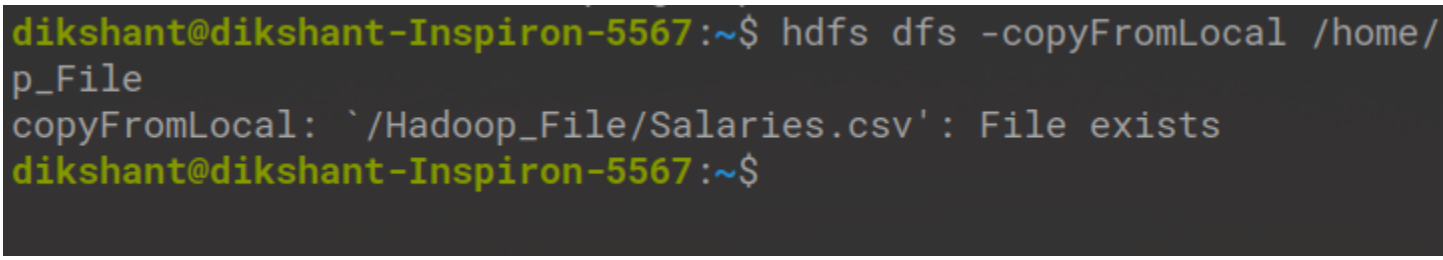
```
hdfs dfs -ls /Hadoop_File
```

```
dikshant@dikshant-Inspiron-5567:~$ hdfs dfs -ls /Hadoop_File
Found 1 items
-rw-r--r--   1 dikshant supergroup       2190 2020-06-23 13:07 /H
dikshant@dikshant-Inspiron-5567:~$
```



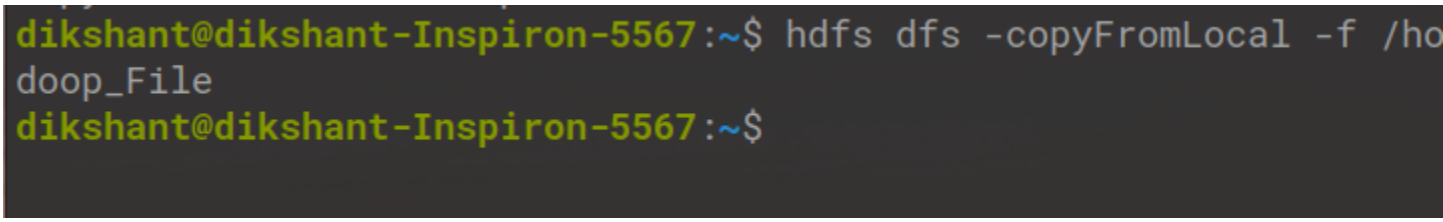
Overwriting or Updating the File In HDFS with *-f* switch

From below Image, you can observe that *copyFromLocal* command itself does not copy the same name file at the same location. it says that the file already exists.



To update the content of the file or to Overwrite it, you should use *-f* switch as shown below.

```
hdfs dfs -copyFromLocal -f
/home/dikshant/Documents/hadoop_file/Salaries.csv /Hadoop_File
```



Now you can easily observe that using *copyFromLocal* with *-f* switch does not produce any error or it will easily update or modify your file in HDFS.

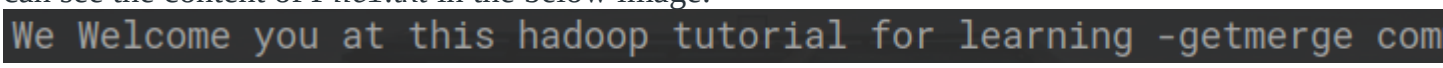
Hadoop – getmerge Command

Hadoop *-getmerge* command is used to merge multiple files in an HDFS(Hadoop Distributed File System) and then put it into one single output file in our local file system.

We want to merge the 2 files present inside are HDFS i.e. *file1.txt* and *file2.txt*, into a single file *output.txt* in our local file system.

Steps To Use -getmerge Command

Step 1: Let's see the content of *file1.txt* and *file2.txt* that are available in our HDFS. You can see the content of *File1.txt* in the below image:



Content of File2.txt

```
getmerge command is used to merge n number of files and then put
ystem
```

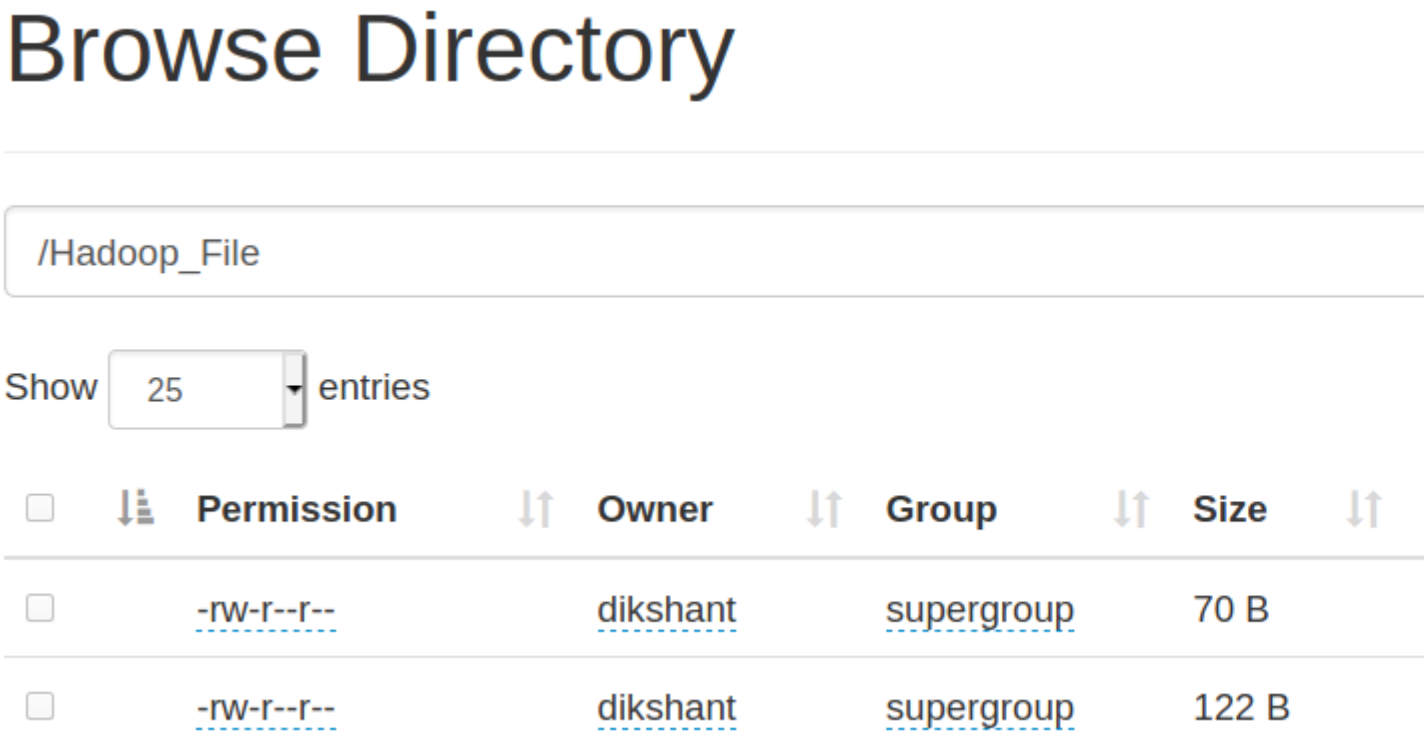
In this case, we have copied both of these files inside my HDFS in *Hadoop_File* folder. If you don't know how to make the directory and copy files to HDFS then follow below command to do so.

- Making Hadoop_Files directory in our HDFS
  `hdfs dfs -mkdir /Hadoop_File`
- Copying files to HDFS

  hdfs dfs -copyFromLocal /home/dikshant/Documents/hadoop_file/file1.txt /home/dikshant/Documents/hadoop_file/file2.txt /Hadoop_File

```
dikshant@dikshant-Inspiron-5567:~$ hdfs dfs -mkdir /Hadoop_File
dikshant@dikshant-Inspiron-5567:~$ hdfs dfs -copyFromLocal /home/
shant/Documents/hadoop_file/file2.txt /Hadoop_File
dikshant@dikshant-Inspiron-5567:~$
```

Below is the Image showing this file inside my /Hadoop_File directory in HDFS.



Step 2: Now it's time to use *-getmerge* command to merge these files into a single output file in our local file system for that follow the below procedure.
Syntax:

```
hdfs dfs -getmerge -nl /path1 /path2 ..../path n /destination
```

-nl is used for adding new line. this will add a new line between the content of these n files. In this case we have merge it to */hadoop_file* folder inside my */Documents* folder.
hdfs dfs -getmerge -nl /Hadoop_File/file1.txt /Hadoop_File/file2.txt /home/dikshant/Documents/hadoop_file/output.txt

```
dikshant@dikshant-Inspiron-5567:~$ hdfs dfs -getmerge -nl /Hadoop
ant/Documents/hadoop_file/output.txt
dikshant@dikshant-Inspiron-5567:~$
```

Now let's see whether the file get merged in *output.txt* file or not.

```
dikshant@dikshant-Inspiron-5567:~$ cd /home/dikshant/Documents/ha
dikshant@dikshant-Inspiron-5567:~/Documents/hadoop_file$ ls
file1.txt  file2.txt  output.txt
dikshant@dikshant-Inspiron-5567:~/Documents/hadoop_file$ cat outp
We Welcome you at this hadoop tutorial for learning -getmerge com

getmerge command is used to merge n number of files and then put
ystem

dikshant@dikshant-Inspiron-5567:~/Documents/hadoop_file$
```

In the above image, you can easily see that the file is merged successfully in our *output.txt* file.