Experiment 2: To implement Various Hadoop HDFS Commands

What is Hadoop?

Apache Hadoop is an open source software framework used to develop data processing applications which are executed in a distributed computing environment.

Applications built using HADOOP are run on large data sets distributed across clusters of commodity computers. Commodity computers are cheap and widely available. These are mainly useful for achieving greater computational power at low cost.

Similar to data residing in a local file system of a personal computer system, in Hadoop, data resides in a distributed file system which is called as a **Hadoop Distributed File system**. The processing model is based on **'Data Locality'** concept wherein computational logic is sent to cluster nodes(server) containing data. This computational logic is nothing, but a compiled version of a program written in a high-level language such as Java. Such a program, processes data stored in Hadoop HDFS.

Hadoop Distributed File System (HDFS) - Data Storage and Management

This is the most important component of the Hadoop ecosystem. HDFS is Hadoop's primary storage system. Hadoop Distributed File System (HDFS) is a Java-based file system that provides reliable, fault tolerance and accessible data storage for the big data. HDFS is a distributed file system that runs on conventional hardware. HDFS is already configured with the default settings for many installations. Typically, a large cluster configuration is required. Hadoop interacts directly with HDFS using commands. Wen comes to HDFS, there are also two components can be identified, which are known as **Name Node** and **Data Node**.

Name Node

It is also known as Master node. Here, it does not store actual data or datasets. Name Node stores the Meta data, for an example, the number of calls transform from a tower, their position, where the end users are getting the call, the Data node data and other details. Basically, this contains files and directories. The tasks of Name node can be recognized as follows.

- Managing file system namespace
- Controlling the access of clients to files
- Executing file system through naming, opening, closing files and directories

Data Node

Data node is called as Slave. Data node is responsible for the effective storage of data in HDFS. The data node completes read and write operations on customer request. They also send signals, known as heartbeats, to the name node. These heartbeats show the status of the data node. Replica Block of Data node consists of two files in the file system. The first file is for data and the second for registry metadata. HDFS metadata contains a data control. At startup, each Data node is connected to the appropriate Name node and grasp. The ID of the Data Node namespace and the software version are controlled by the handshake. If a discrepancy is detected, Data Node is automatically disabled. When comes to tasks of Data node, those can be detailed as follows.

- This is consisting of operations like block replica creation, deletion, and replication according to the instruction of Name node
- Managing data storage of the system

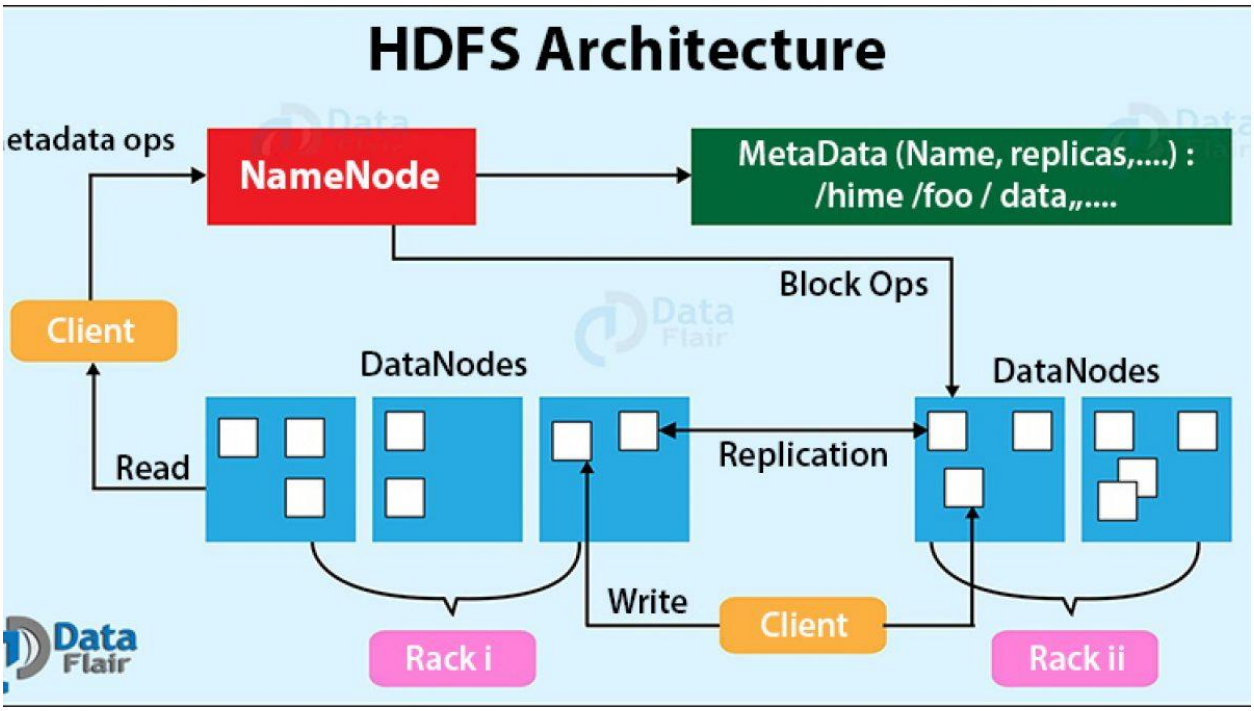
Processing and Computation – Hadoop MapReduce

When comes to Hadoop MapReduce, that is the main component of the Hadoop, that provides data processing. MapReduce is can be identified as an easy-to-write application framework that processes the large amount of structured and unstructured data stored in the Hadoop distributed file system.

MapReduce programs are parallel, so they are very useful for large-scale data analysis using multiple clusters. Therefore, this parallelism increases the speed and reliability of the cluster. In MapReduce, there are two functions, Map function and Reduce function.

Two functions can be identified, map function and reduce function.

- The map function retrieves a data set and converts it to another data set. Each element is divided into processing (key / value pairs).
- The Reduce function accepts the Map output as an input and integrates these data nodes based on the key and changes the key value accordingly.



Hadoop HDFS Commands

1) Hadoop Version

->hadoop version

The Hadoop fs shell command **version** prints the Hadoop version.

```
File  Edit  View  Search  Terminal  Help
[cloudera@quickstart ~]$ hadoop version
Hadoop 2.6.0-cdh5.13.0
Subversion http://github.com/cloudera/hadoop -r 42e8860b182e55321bd5f5605264da4adc8882be
Compiled by jenkins on 2017-10-04T18:08Z
Compiled with protoc 2.5.0
From source with checksum 5e84c185f8a22158e2b0e4b8f85311
This command was run using /usr/lib/hadoop/hadoop-common-2.6.0-cdh5.13.0.jar
[cloudera@quickstart ~]$
```

2) LS Command

->hdfs dfs -ls /

HDFS Command to display the list of Files and Directories in HDFS. It Lists the contents of the directory specified by path, showing the names, permissions, owner, size and modification date for each entry.

hdfs dfs is the command that is specific to HDFS.

```
[cloudera@quickstart ~]$ hdfs dfs -ls /
Found 10 items
drwxrwxrwx   - hdfs     supergroup          0 2017-10-23 09:15 /benchmarks
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:58 /forcopy
drwxr-xr-x   - hbase    supergroup          0 2021-06-04 07:57 /hbase
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 00:06 /inputdir
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 06:34 /op1
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 00:37 /outputdir
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:55 /solr
drwxrwxrwt   - hdfs     supergroup          0 2021-05-24 10:39 /tmp
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /user
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /var
[cloudera@quickstart ~]$
```

->hadoop fs -ls /

hadoop fs is more "generic" command that allows you to interact with multiple file systems including Hadoop. we are using the **ls** command to enlist the files and directories present in HDFS. The Hadoop fs shell command ls displays a list of the contents of a directory specified in the path provided by the user. It shows the name, permissions, owner, size, and modification date for each file or directories in the specified directory.

Using the **ls** command, we can check for the directories in HDFS.

```
[cloudera@quickstart ~]$ hadoop fs -ls /
Found 10 items
drwxrwxrwx   - hdfs     supergroup          0 2017-10-23 09:15 /benchmarks
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:58 /forcopy
drwxr-xr-x   - hbase    supergroup          0 2021-06-04 07:57 /hbase
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 00:06 /inputdir
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 06:34 /op1
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 00:37 /outputdir
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:55 /solr
drwxrwxrwt   - hdfs     supergroup          0 2021-05-24 10:39 /tmp
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /user
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /var
[cloudera@quickstart ~]$
```

2) MKDIR Command

HDFS Command to create the directory in HDFS.

Usage: hdfs dfs –mkdir /directory_name

Here I am trying to create a directory named "rjc" in HDFS.

->hdfs dfs -mkdir /rjc

```
[cloudera@quickstart ~]$ hdfs dfs -mkdir /rjc
[cloudera@quickstart ~]$
```

Using the ls command, we can check for the directories in HDFS or Using ls command we listed the directory ‘rjc” created using mkdir

```
[cloudera@quickstart ~]$ hdfs dfs -ls /
Found 11 items
drwxrwxrwx   - hdfs     supergroup          0 2017-10-23 09:15 /benchmarks
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:58 /forcopy
drwxr-xr-x   - hbase    supergroup          0 2021-06-04 07:57 /hbase
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 00:06 /inputdir
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 06:34 /op1
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 00:37 /outputdir
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 12:19 /rjc
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:55 /solr
drwxrwxrwt   - hdfs     supergroup          0 2021-05-24 10:39 /tmp
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /user
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /var
```

3) ->Hadoop dfsadmin -safemode leave

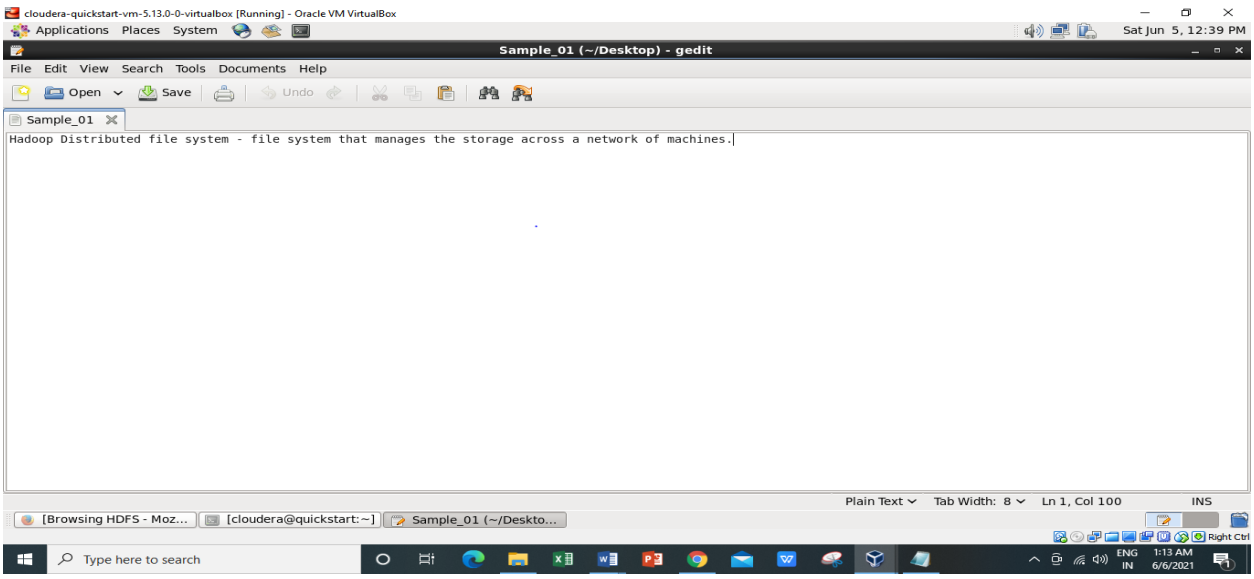
Use this command if you are getting “you are in safe mode “

4) <u>copyFromLocal Command</u>

First we will Create a document.

Steps: Right click anywhere on desktop->empty file->Sample_01

Put some information in the file Sample_01.



Now we are trying to copy the ‘Sample_01’ file present in the local file system to the ‘rjc’ directory of Hadoop. Below command copies the file from the local file system to HDFS.

->hdfs dfs -copyFromLocal /home/cloudera/Desktop/Sample_01 /rjc

```
[cloudera@quickstart ~]$ hdfs dfs -copyFromLocal /home/cloudera/Desktop/Sample_01 /rjc
[cloudera@quickstart ~]$
```

Now Using the ls command, we can check for the directories in HDFS.

```
[cloudera@quickstart ~]$ hdfs dfs -copyFromLocal /home/cloudera/Desktop/Sample_01 /rjc
[cloudera@quickstart ~]$ hdfs dfs -ls /
Found 11 items
drwxrwxrwx   - hdfs     supergroup          0 2017-10-23 09:15 /benchmarks
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:58 /forcopy
drwxr-xr-x   - hbase    supergroup          0 2021-06-04 07:57 /hbase
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 00:06 /inputdir
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 06:34 /op1
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 00:37 /outputdir
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 12:49 /rjc
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:55 /solr
drwxrwxrwt   - hdfs     supergroup          0 2021-05-24 10:39 /tmp
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /user
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /var
[cloudera@quickstart ~]$
```

5) If getting any error due to permissions

Use-> export HADOOP_USER_NAME=hdfs

6) <u>Put Command</u>

-> hdfs dfs -put /home/cloudera/Desktop/Sample_01 /rjc

Here in this example, we are trying to copy “Sample_01” of the local file system to the Hadoop filesystem.

The Hadoop fs shell command **put** is similar to the **copyFromLocal**, which copies files or directory from the local filesystem to the destination in the Hadoop filesystem

```
[cloudera@quickstart ~]$ hadoop fs -put /home/cloudera/Desktop/Sample_01 /rjc
put: `/rjc/Sample_01': File exists
[cloudera@quickstart ~]$
```

Now Using the ls command, we can check for the directories in HDFS.

```
[cloudera@quickstart ~]$ hdfs dfs -ls /
Found 11 items
drwxrwxrwx   - hdfs     supergroup          0 2017-10-23 09:15 /benchmarks
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:58 /forcopy
drwxr-xr-x   - hbase    supergroup          0 2021-06-04 07:57 /hbase
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 00:06 /inputdir
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 06:34 /op1
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 00:37 /outputdir
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 12:49 /rjc
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:55 /solr
drwxrwxrwt   - hdfs     supergroup          0 2021-05-24 10:39 /tmp
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /user
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /var
[cloudera@quickstart ~]$
```

7) copyToLocal Command

->**hdfs dfs -copyToLocal /rjc/Sample /home/cloudera/Desktop**

copyToLocal command copies the file from HDFS to the local file system

Here in this example, we are trying to copy the 'Sample' file present in the rjc directory of HDFS to the local file system.

Deleted Sample file from desktop. If it is already exist. And then again run the command.

```
[cloudera@quickstart ~]$ hdfs dfs -ls /rjc
Found 1 items
-rw-r--r--   1 cloudera supergroup        100 2021-06-05 12:49 /rjc/Sample_01
[cloudera@quickstart ~]$ hdfs dfs -copyFromLocal /home/cloudera/Desktop/Sample /rjc
[cloudera@quickstart ~]$ hdfs dfs -ls /
Found 11 items
drwxrwxrwx   - hdfs     supergroup          0 2017-10-23 09:15 /benchmarks
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:58 /forcopy
drwxr-xr-x   - hbase    supergroup          0 2021-06-06 01:08 /hbase
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 00:06 /inputdir
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 06:34 /op1
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 00:37 /outputdir
drwxr-xr-x   - cloudera supergroup          0 2021-06-06 01:49 /rjc
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:55 /solr
drwxrwxrwt   - hdfs     supergroup          0 2021-05-24 10:39 /tmp
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /user
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /var
[cloudera@quickstart ~]$ hdfs dfs -ls /rjc
Found 2 items
-rw-r--r--   1 cloudera supergroup        155 2021-06-06 01:49 /rjc/Sample
-rw-r--r--   1 cloudera supergroup        100 2021-06-05 12:49 /rjc/Sample_01
[cloudera@quickstart ~]$ hdfs dfs -copyToLocal /rjc/Sample /home/cloudera/Desktop
copyToLocal: `/home/cloudera/Desktop/Sample': File exists
[cloudera@quickstart ~]$ hdfs dfs -copyToLocal /rjc/Sample /home/cloudera/Desktop
copyToLocal: `/home/cloudera/Desktop/Sample': File exists
[cloudera@quickstart ~]$ hdfs dfs -copyToLocal /rjc/Sample /home/cloudera/Desktop
```

8) CAT Command

->hdfs dfs -cat /rjc/Sample_01

we are using the cat command to display the content of the ‘Sample_01’ file present in rjc directory of HDFS

The **cat** command reads the file in HDFS and displays the content of the file on console or stdout.

```
[cloudera@quickstart ~]$ hdfs dfs -cat /rjc/Sample_01
Hadoop Distributed file system - file system that manages the storage across a network of machines.
[cloudera@quickstart ~]$
```

9) Cp Command

First we will create ‘newdir’ inside hdfs and then we copy ‘Sample’ file which is present in ‘rjc’ folder inside this ‘newdir’ directory in hdfs.

->hdfs dfs -cp /rjc/Sample /newdir

we are copying the ‘sample’ present in rjc directory in HDFS to the newdir of HDFS.

The **cp** command copies a file from one directory to another directory within the HDFS.

```
[cloudera@quickstart ~]$ hdfs dfs -mkdir /newdir
[cloudera@quickstart ~]$ hdfs dfs -ls /newdir
[cloudera@quickstart ~]$ hdfs dfs -cp /rjc/Sample /newdir
[cloudera@quickstart ~]$ hdfs dfs -ls /newdir
Found 1 items
-rw-r--r--   1 cloudera supergroup        155 2021-06-06 02:14 /newdir/Sample
[cloudera@quickstart ~]$
```

10) MV Command

->hdfs dfs -mv /rjc/Sample_01/output_new_2

we have a directory ‘rjc’ in HDFS. We are using **mv** command to move the rjc directory to the output_new directory in HDFS.

The HDFS mv command moves the files or directories from the source to a destination within HDFS**.**

```
[cloudera@quickstart ~]$ hdfs dfs -mv /rjc/Sample_01 /output_new_2
[cloudera@quickstart ~]$ hdfs dfs -ls /output_new_2
-rw-r--r--   1 cloudera supergroup        100 2021-06-06 02:32 /output_new_2
[cloudera@quickstart ~]$ hdfs dfs -cat /output_new_2
Hadoop Distributed file system - file system that manages the storage across a network of machines.
[cloudera@quickstart ~]$
```

11) RM Command

->hdfs dfs –rm /output_3/Sample

The hadoop dfs -rm command deletes objects and directories full of objects.

```
[cloudera@quickstart ~]$ hdfs dfs -mkdir /output_3
[cloudera@quickstart ~]$ hdfs dfs -ls /
Found 14 items
drwxrwxrwx   - hdfs     supergroup          0 2017-10-23 09:15 /benchmarks
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:58 /forcopy
drwxr-xr-x   - hbase    supergroup          0 2021-06-06 01:08 /hbase
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 00:06 /inputdir
drwxr-xr-x   - cloudera supergroup          0 2021-06-06 02:14 /newdir
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 06:34 /op1
drwxr-xr-x   - cloudera supergroup          0 2021-06-06 02:46 /output_3
drwxr-xr-x   - cloudera supergroup          0 2021-06-06 02:28 /output_new
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 00:37 /outputdir
drwxr-xr-x   - cloudera supergroup          0 2021-06-06 02:33 /rjc
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:55 /solr
drwxrwxrwt   - hdfs     supergroup          0 2021-05-24 10:39 /tmp
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /user
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /var
[cloudera@quickstart ~]$ ^C
[cloudera@quickstart ~]$ hdfs dfs -cp /rjc/Sample /output_3
[cloudera@quickstart ~]$ hdfs dfs -ls /output_3
Found 1 items
-rw-r--r--   1 cloudera supergroup        155 2021-06-06 02:48 /output_3/Sample
[cloudera@quickstart ~]$ hdfs dfs -rm /output_3/Sample
Deleted /output_3/Sample
[cloudera@quickstart ~]$ hdfs dfs -ls /output_3
[cloudera@quickstart ~]$
```

->hdfs dfs -rm -r /output_3

In case, we want to delete a directory which contains files, `-rm` will not be able to delete the directory. In that case we can use recursive option for removing all the files from the directory following by removing the directory when it is empty.

```
[cloudera@quickstart ~]$ hdfs dfs -rm -r /output_3
Deleted /output_3
[cloudera@quickstart ~]$ hdfs dfs -ls /
Found 13 items
drwxrwxrwx   - hdfs     supergroup          0 2017-10-23 09:15 /benchmarks
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:58 /forcopy
drwxr-xr-x   - hbase    supergroup          0 2021-06-06 01:08 /hbase
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 00:06 /inputdir
drwxr-xr-x   - cloudera supergroup          0 2021-06-06 02:14 /newdir
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 06:34 /op1
drwxr-xr-x   - cloudera supergroup          0 2021-06-06 02:28 /output_new
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 00:37 /outputdir
drwxr-xr-x   - cloudera supergroup          0 2021-06-06 02:33 /rjc
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:55 /solr
drwxrwxrwt   - hdfs     supergroup          0 2021-05-24 10:39 /tmp
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /user
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /var
```

12) MoveFromLocal Command

->hdfs dfs -moveFromLocal /home/cloudera/Desktop/new /output_abc

The Hadoop fs shell command **moveFromLocal** moves the file or directory from the local filesystem to the destination in Hadoop HDFS.

```
[cloudera@quickstart ~]$ ls /home/cloudera/Desktop
abc   BigData-Practical01   BigData-Practical-03-WordCount  Enterprise.desktop  Kerberos.desktop  new~            Sample   Sample_01   Sample - 1~
abc~  BigData-Practical01~  Eclipse.desktop                 Express.desktop     new               Parcels.desktop  Sample~  Sample_01~
[cloudera@quickstart ~]$ hdfs dfs -moveFromLocal /home/cloudera/Desktop/new /output_abc
[cloudera@quickstart ~]$ hdfs df -ls /
Error: Could not find or load main class df
[cloudera@quickstart ~]$ hdfs dfs -ls /
Found 14 items
drwxrwxrwx   - hdfs     supergroup          0 2017-10-23 09:15 /benchmarks
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:58 /forcopy
drwxr-xr-x   - hbase    supergroup          0 2021-06-06 01:08 /hbase
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 00:06 /inputdir
drwxr-xr-x   - cloudera supergroup          0 2021-06-06 02:14 /newdir
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 06:34 /op1
-rw-r--r--   1 cloudera supergroup        155 2021-06-06 03:17 /output_abc
drwxr-xr-x   - cloudera supergroup          0 2021-06-06 02:28 /output_new
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 00:37 /outputdir
drwxr-xr-x   - cloudera supergroup          0 2021-06-06 02:33 /rjc
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:55 /solr
drwxrwxrwt   - hdfs     supergroup          0 2021-05-24 10:39 /tmp
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /user
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /var
```

13) MoveToLocal Command

->hdfs dfs -moveToLocal /new /home/cloudera/Desktop/new

The Hadoop fs shell command moveToLocal moves the file or directory from the Hadoop filesystem to the destination in the local filesystem.

This command is not yet implemented by Hadoop

```
[cloudera@quickstart ~]$ hdfs dfs -moveToLocal  /new /home/cloudera/Desktop/new
moveToLocal: Option '-moveToLocal' is not implemented yet.
[cloudera@quickstart ~]$
```

14) Tail Command

->hdfs dfs -tail /test_1

The Hadoop fs shell **tail** command shows the last 1KB of a file on console or stdout.

```
[cloudera@quickstart ~]$ hdfs dfs -moveFromLocal /home/cloudera/Desktop/test /test_1
[cloudera@quickstart ~]$ hdfs dfs -ls /
Found 15 items
drwxrwxrwx   - hdfs     supergroup          0 2017-10-23 09:15 /benchmarks
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:58 /forcopy
drwxr-xr-x   - hbase    supergroup          0 2021-06-06 01:08 /hbase
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 00:06 /inputdir
drwxr-xr-x   - cloudera supergroup          0 2021-06-06 02:14 /newdir
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 06:34 /op1
-rw-r--r--   1 cloudera supergroup        155 2021-06-06 03:17 /output_abc
drwxr-xr-x   - cloudera supergroup          0 2021-06-06 02:28 /output_new
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 00:37 /outputdir
drwxr-xr-x   - cloudera supergroup          0 2021-06-06 02:33 /rjc
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:55 /solr
-rw-r--r--   1 cloudera supergroup       4049 2021-06-06 03:53 /test_1
drwxrwxrwt   - hdfs     supergroup          0 2021-05-24 10:39 /tmp
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /user
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /var
[cloudera@quickstart ~]$ hdfs dfs -tail /test_1
truction of Name node
•       Managing data storage of the system
Processing and Computation – Hadoop MapReduce
When comes to Hadoop MapReduce, that is the main component of the Hadoop, that provides data processing. MapReduce is can be identified as an easy-to-write application
framework that processes the large amount of structured and unstructured data stored in the Hadoop distributed file system.
MapReduce programs are parallel, so they are very useful for large-scale data analysis using multiple clusters. Therefore, this parallelism increases the speed and reli
ability of the cluster. In MapReduce, there are two functions, Map function and Reduce function.
Two functions can be identified, map function and reduce function.
•       The map function retrieves a data set and converts it to another data set. Each element is divided into processing (key / value pairs).
•       The Reduce function accepts the Map output as an input and integrates these data nodes based on the key and changes the key value accordingly.

[cloudera@quickstart ~]$
```

15) Expunge Command

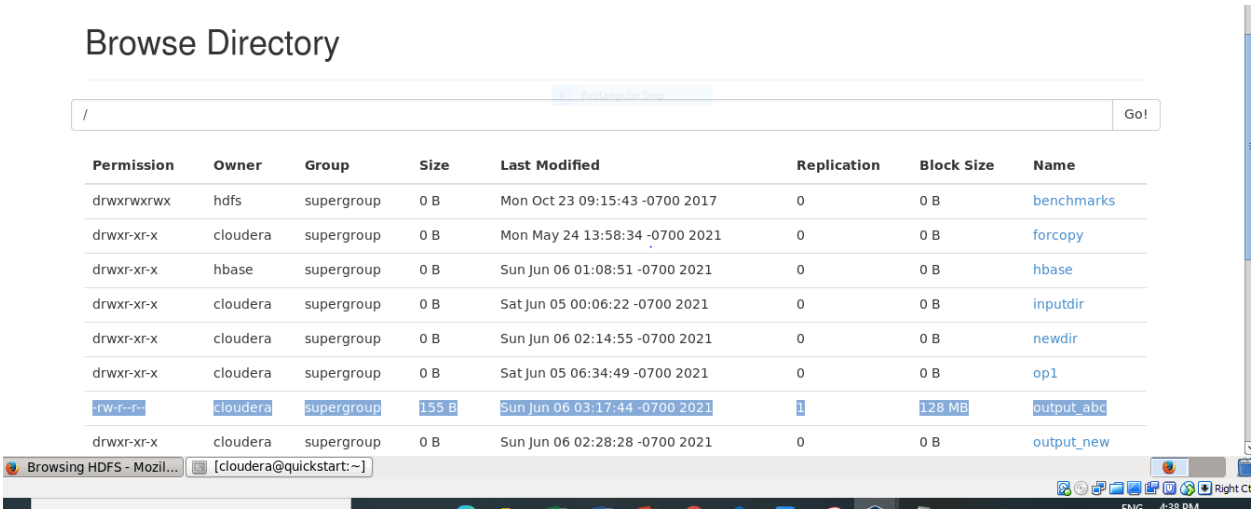
->hdfs dfs -expunge

This command is used to empty the trash available in an **HDFS** system

```
[cloudera@quickstart ~]$ hdfs dfs  -expunge
[cloudera@quickstart ~]$
```

16) Replication Command

Earlier Replication Number of output_abc is 1



->hdfs dfs -setrep 4 /output_abc

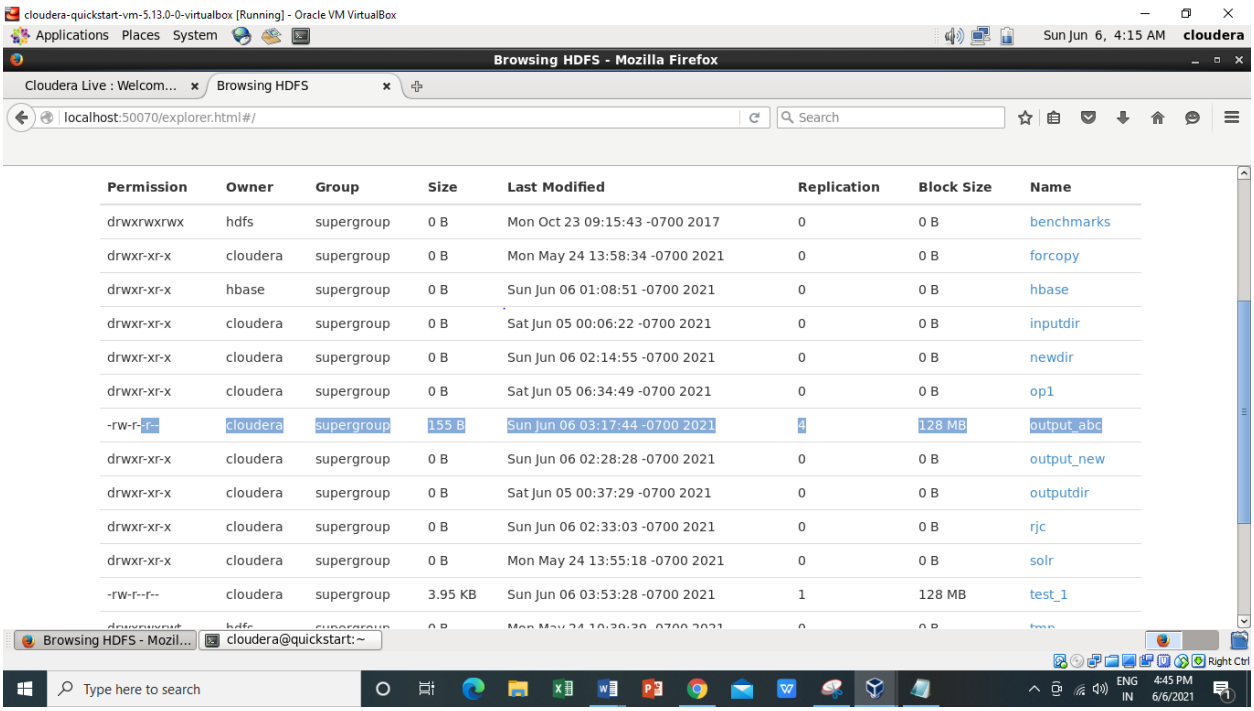
This command is used to change the replication factor of a file to a specific count instead of the default replication factor for the remaining in the HDFS file system

```
[cloudera@quickstart ~]$ hdfs dfs -setrep 4 /output_abc
Replication 4 set: /output_abc
[cloudera@quickstart ~]$
```

Now Replication factor is 4

-rw-r--r--	cloudera	supergroup	155 B	Sun Jun 06 03:17:44 -0700 2021	4	128 MB	output_abc

Or



17) DU Command

->hdfs dfs -du /rjc

Use the hdfs du command to get the size of a directory in HDFS

du stands for disk usage.

```
[cloudera@quickstart ~]$ hdfs dfs -du /rjc
155  155  /rjc/Sample
[cloudera@quickstart ~]$
```

Since the replication factor of output_abc file was set as 4 earlier we could see that 19 and 155*4=620

->hdfs dfs -du /output_abc

```
[cloudera@quickstart ~]$ hdfs dfs -du /output_abc
155  620  /output_abc
[cloudera@quickstart ~]$
```

18) Df command

->hdfs dfs -df

To get all the space related details of the Hadoop File System we can use `df` command. It provides the information regarding the amount of space used and amount of space available on the currently mounted filesystem

```
[cloudera@quickstart ~]$ hdfs dfs -df
Filesystem                              Size       Used    Available  Use%
hdfs://quickstart.cloudera:8020  58531520512  873140224  45726425088    1%
[cloudera@quickstart ~]$
```

->hdfs dfs -df -h

With h parameter the information is human readable

```
[cloudera@quickstart ~]$ hdfs dfs -df -h
Filesystem                          Size     Used  Available  Use%
hdfs://quickstart.cloudera:8020  54.5 G  832.7 M     42.6 G    1%
[cloudera@quickstart ~]$
```

19) Fsck command

The fsck Hadoop command is used to check the health of the HDFS. It moves a corrupted file to the lost+found directory. It deletes the corrupted files present in HDFS. It prints the files being checked.

->hdfs fsck /rjc

```
[cloudera@quickstart ~]$ hdfs fsck /rjc
Connecting to namenode via http://quickstart.cloudera:50070/fsck?ugi=cloudera&path=%2Frjc
FSCK started by cloudera (auth:SIMPLE) from /10.0.2.15 for path /rjc at Sun Jun 06 04:26:35 PDT 2021
.Status: HEALTHY
 Total size:    155 B
 Total dirs:    1
 Total files:   1
 Total symlinks:                0
 Total blocks (validated):      1 (avg. block size 155 B)
 Minimally replicated blocks:   1 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       0 (0.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    1
 Average block replication:     1.0
 Corrupt blocks:                0
 Missing replicas:              0 (0.0 %)
 Number of data-nodes:          1
 Number of racks:               1
FSCK ended at Sun Jun 06 04:26:36 PDT 2021 in 993 milliseconds


The filesystem under path '/rjc' is HEALTHY
[cloudera@quickstart ~]$
```

->hdfs fsck /rjc -files

```
[cloudera@quickstart ~]$ hdfs fsck /rjc -files
Connecting to namenode via http://quickstart.cloudera:50070/fsck?ugi=cloudera&files=1&path=%2Frjc
FSCK started by cloudera (auth:SIMPLE) from /10.0.2.15 for path /rjc at Sun Jun 06 04:28:52 PDT 2021
/rjc <dir>
/rjc/Sample 155 bytes, 1 block(s):  OK
Status: HEALTHY
 Total size:    155 B
 Total dirs:    1
 Total files:   1
 Total symlinks:                0
 Total blocks (validated):      1 (avg. block size 155 B)
 Minimally replicated blocks:   1 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       0 (0.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    1
 Average block replication:     1.0
 Corrupt blocks:                0
 Missing replicas:              0 (0.0 %)
 Number of data-nodes:          1
 Number of racks:               1
FSCK ended at Sun Jun 06 04:28:52 PDT 2021 in 3 milliseconds


The filesystem under path '/rjc' is HEALTHY
[cloudera@quickstart ~]$
```

20) Touchz Command

It creates an empty file.

->hdfs dfs -touchz /rjc/file1

```
[cloudera@quickstart ~]$ hdfs dfs -touchz /rjc/file1
[cloudera@quickstart ~]$ hdfs dfs -ls /rjc
Found 2 items
-rw-r--r--   1 cloudera supergroup        155 2021-06-06 01:49 /rjc/Sample
-rw-r--r--   1 cloudera supergroup          0 2021-06-06 04:32 /rjc/file1
[cloudera@quickstart ~]$
```

21) Stat Command

The Hadoop fs shell command stat prints the statistics about the file or directory in the specified format

It shows the recent date of modification.

->hdfs dfs -stat /rjc

```
[cloudera@quickstart ~]$ hdfs dfs -stat /rjc
2021-06-06 11:32:33
[cloudera@quickstart ~]$
```

22) ->hdfs dfs -stat %b /rjc/file1

%b shows byte size of file

```
[cloudera@quickstart ~]$ hdfs dfs -stat %b /rjc/file1
0
[cloudera@quickstart ~]$
```

23) ->hdfs dfs -stat %o /output_abc

%o gives size of block in bytes

```
[cloudera@quickstart ~]$ hdfs dfs -stat %o /output_abc
134217728
[cloudera@quickstart ~]$
```

24) ->hdfs dfs -stat %r /output_abc

% r gives replication factor

```
[cloudera@quickstart ~]$ hdfs dfs -stat %r /output_abc
4
[cloudera@quickstart ~]$
```

25) ->hdfs dfs -stat %y /output_abc

% y gives modification date

```
[cloudera@quickstart ~]$ hdfs dfs -stat %y /output_abc
2021-06-06 10:17:44
[cloudera@quickstart ~]$
```

26) Checksum Command

checksum property, which defaults to 512 bytes. The chunk size is stored as metadata in the . crc file, so the file can be read back correctly even if the setting for the chunk size has change

->hdfs dfs -checksum /output_abc

```
[cloudera@quickstart ~]$ hdfs dfs -checksum /output_abc
/output_abc     MD5-of-0MD5-of-512CRC32C        000002000000000000000000f64588c6097799afc7190f4f3c95bfa2
[cloudera@quickstart ~]$
```

27) Help command

->hdfs dfs -help mkdir

Shows the syntax of whereas commands

```
[cloudera@quickstart ~]$ hdfs dfs -help mkdir
-mkdir [-p] <path> ... :
  Create a directory in specified location.

  -p  Do not fail if the directory already exists
[cloudera@quickstart ~]$
```

28) Get command

-> hdfs dfs -get /rjc/Sample /home/cloudera/Desktop/new_1

The Hadoop fs shell command get copies the file or directory from the Hadoop file system to the local file system.

```
[cloudera@quickstart ~]$ hdfs dfs -get /rjc/Sample /home/cloudera/Desktop/new_1
[cloudera@quickstart ~]$ hdfs dfs -cat /rjc/Sample
Apache Hadoop is an open source software framework used to develop data processing applications which are executed in a distributed computing environment.
[cloudera@quickstart ~]$ hdfs dfs -cat /home/cloudera/Desktop/new_1
cat: `/home/cloudera/Desktop/new_1': No such file or directory
[cloudera@quickstart ~]$ hadoop fs -cat /home/cloudera/Desktop/new_1
cat: `/home/cloudera/Desktop/new_1': No such file or directory
[cloudera@quickstart ~]$ ls
cloudera-manager  Desktop    Downloads  enterprise-deployment.json  hdfs      lib    parcels   Public     Videos         workspace
cm_api.py         Documents  eclipse    express-deployment.json     kerberos  Music  Pictures  Templates  WordCount.jar
[cloudera@quickstart ~]$ ls /home/cloudera/Desktop
abc   BigData-Practical01   BigData-Practical-03-WordCount  Enterprise.desktop  Kerberos.desktop  new_1           Sample   Sample_01   Sample - 1~
abc~  BigData-Practical01~  Eclipse.desktop                 Express.desktop     new~              Parcels.desktop  Sample~  Sample_01~  test~
[cloudera@quickstart ~]$
```