

IoT & Applications (18EI2T09)

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
```

```
suraj@suraj:~/hadoop-2.5.0-cdh5.3.2$ jps
2546 SecondaryNameNode
2404 DataNode
2295 NameNode
2760 ResourceManager
2874 NodeManager
4251 Jps
suraj@suraj:~/hadoop-2.5.0-cdh5.3.2$
```

1. version

Hadoop HDFS version Command Usage:

```
version
```

Hadoop HDFS version Command Example:

Before working with HDFS you need to Deploy Hadoop, follow this guide to **Install and configure Hadoop 3**.

```
hadoop version
```

```
dataflair@admin1-All-Series: ~
File Edit View Search Terminal Help
dataflair@admin1-All-Series:~$ hadoop version
Hadoop 3.1.2
Source code repository https://github.com/apache/hadoop.git -r 1019dde65bcf12e05ef48ac71e84550d589e5d9a
Compiled by sunilg on 2019-01-29T01:39Z
Compiled with protoc 2.5.0
From source with checksum 64b8bdd4ca6e77cce75a93eb09ab2a9
This command was run using /home/dataflair/hadoop-3.1.2/share/hadoop/common/hadoop-common-3.1.2.jar
dataflair@admin1-All-Series:~$
```

Hadoop HDFS version Command Description:

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

IoT & Applications (18EI2T09)

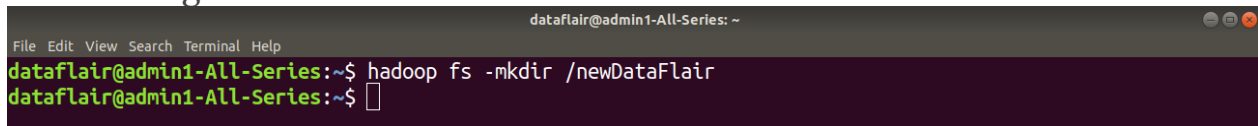
2. mkdir

Hadoop HDFS mkdir Command Usage:

`hadoop fs -mkdir /path/directory_name`

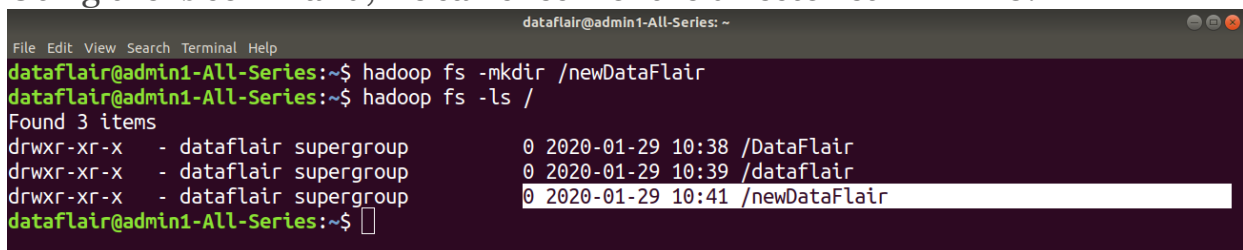
Hadoop HDFS mkdir Command Example 1:

In this example, we are trying to create a new DataFlair named directory in HDFS using the `mkdir` command.



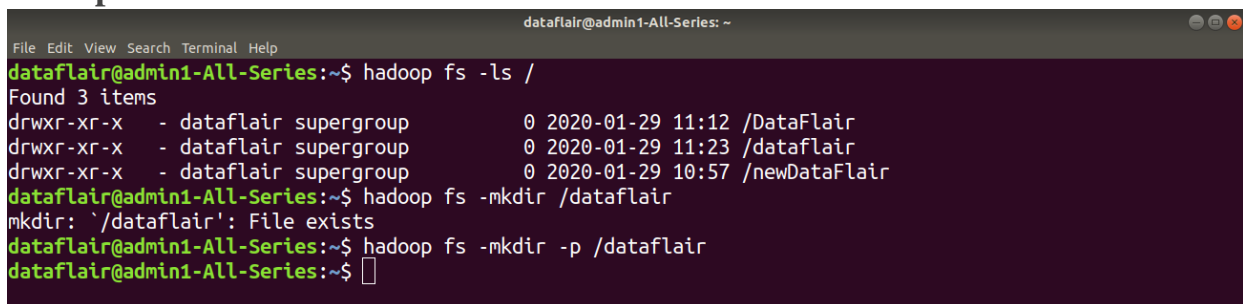
```
dataflair@admin1-All-Series: ~  
File Edit View Search Terminal Help  
dataflair@admin1-All-Series:~$ hadoop fs -mkdir /newDataFlair  
dataflair@admin1-All-Series:~$
```

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



```
dataflair@admin1-All-Series: ~  
File Edit View Search Terminal Help  
dataflair@admin1-All-Series:~$ hadoop fs -mkdir /newDataFlair  
dataflair@admin1-All-Series:~$ hadoop fs -ls /  
Found 3 items  
drwxr-xr-x - dataflair supergroup          0 2020-01-29 10:38 /DataFlair  
drwxr-xr-x - dataflair supergroup          0 2020-01-29 10:39 /dataflair  
drwxr-xr-x - dataflair supergroup          0 2020-01-29 10:41 /newDataFlair  
dataflair@admin1-All-Series:~$
```

Example 2:



```
dataflair@admin1-All-Series: ~  
File Edit View Search Terminal Help  
dataflair@admin1-All-Series:~$ hadoop fs -ls /  
Found 3 items  
drwxr-xr-x - dataflair supergroup          0 2020-01-29 11:12 /DataFlair  
drwxr-xr-x - dataflair supergroup          0 2020-01-29 11:23 /dataflair  
drwxr-xr-x - dataflair supergroup          0 2020-01-29 10:57 /newDataFlair  
dataflair@admin1-All-Series:~$ hadoop fs -mkdir /dataflair  
mkdir: '/dataflair': File exists  
dataflair@admin1-All-Series:~$ hadoop fs -mkdir -p /dataflair  
dataflair@admin1-All-Series:~$
```

Hadoop HDFS mkdir Command Description:

This command creates the directory in HDFS if it does not already exist.

Note: If the directory already exists in HDFS, then we will get an error message that file already exists.

Use `hadoop fs mkdir -p /path/directoryname`, so not to fail even if directory exists.

IoT & Applications (18EI2T09)

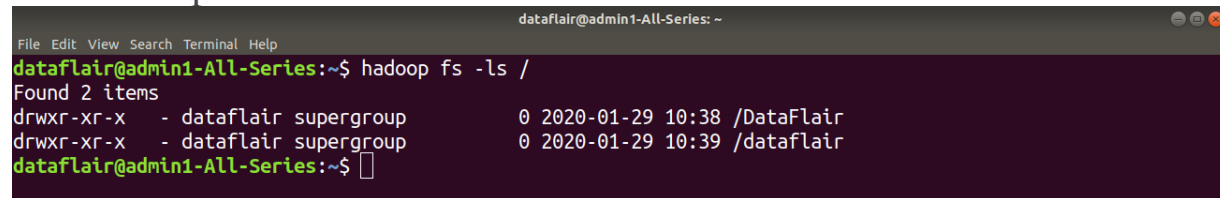
3. ls

Hadoop HDFS ls Command Usage:

`hadoop fs -ls /path`

Hadoop HDFS ls Command Example 1:

Here in the below example, we are using the `ls` command to enlist the files and directories present in HDFS.

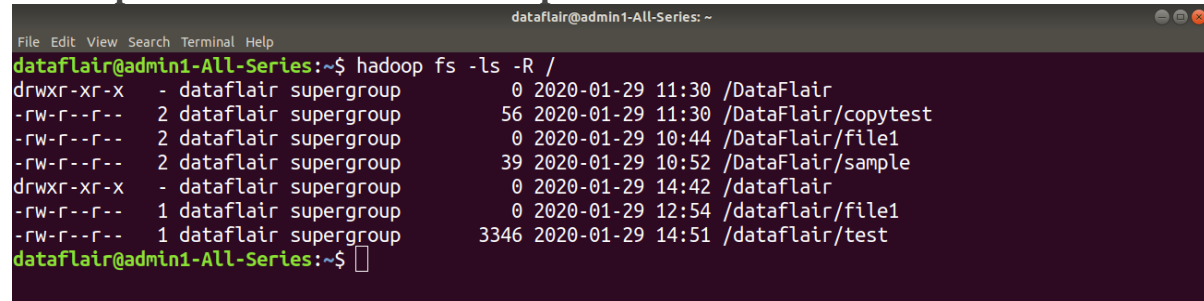


```
dataflair@admin1-All-Series: ~  
File Edit View Search Terminal Help  
dataflair@admin1-All-Series:~$ hadoop fs -ls /  
Found 2 items  
drwxr-xr-x - dataflair supergroup          0 2020-01-29 10:38 /DataFlair  
drwxr-xr-x - dataflair supergroup          0 2020-01-29 10:39 /dataflair  
dataflair@admin1-All-Series:~$
```

Hadoop HDFS ls Command Description:

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.

Hadoop HDFS ls Command Example 2:



```
dataflair@admin1-All-Series: ~  
File Edit View Search Terminal Help  
dataflair@admin1-All-Series:~$ hadoop fs -ls -R /  
drwxr-xr-x - dataflair supergroup          0 2020-01-29 11:30 /DataFlair  
-rw-r--r-- 2 dataflair supergroup          56 2020-01-29 11:30 /DataFlair/copytest  
-rw-r--r-- 2 dataflair supergroup          0 2020-01-29 10:44 /DataFlair/file1  
-rw-r--r-- 2 dataflair supergroup          39 2020-01-29 10:52 /DataFlair/sample  
drwxr-xr-x - dataflair supergroup          0 2020-01-29 14:42 /dataflair  
-rw-r--r-- 1 dataflair supergroup          0 2020-01-29 12:54 /dataflair/file1  
-rw-r--r-- 1 dataflair supergroup       3346 2020-01-29 14:51 /dataflair/test  
dataflair@admin1-All-Series:~$
```

Hadoop HDFS ls Description:

This Hadoop `fs` command behaves like `-ls`, but recursively displays entries in all subdirectories of a path.

4. put

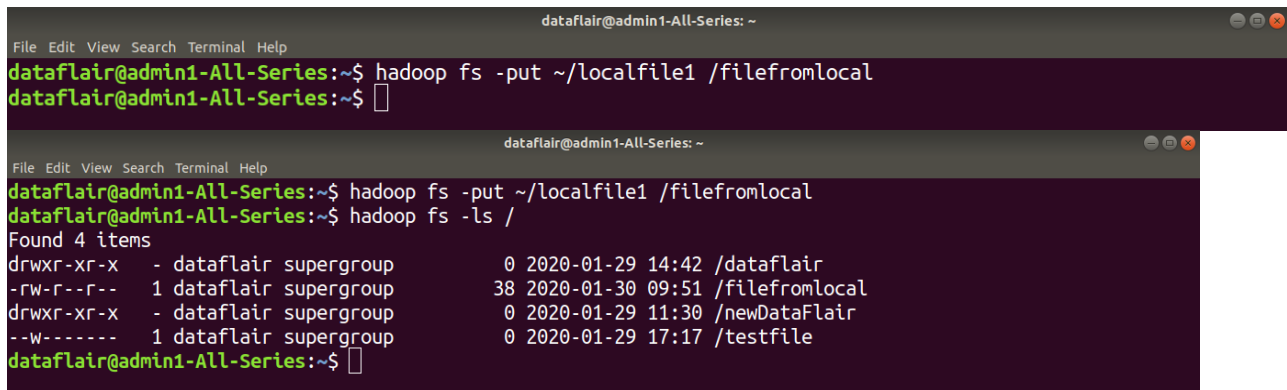
Hadoop HDFS put Command Usage:

`hadoop fs -put <localsrc> <dest>`

Hadoop HDFS put Command Example:

Here in this example, we are trying to copy `localfile1` of the local file system to the Hadoop filesystem.

IoT & Applications (18EI2T09)



```
dataflair@admin1-All-Series: ~  
File Edit View Search Terminal Help  
dataflair@admin1-All-Series:~$ hadoop fs -put ~/localfile1 /filefromlocal  
dataflair@admin1-All-Series:~$  
  
dataflair@admin1-All-Series: ~  
File Edit View Search Terminal Help  
dataflair@admin1-All-Series:~$ hadoop fs -put ~/localfile1 /filefromlocal  
dataflair@admin1-All-Series:~$ hadoop fs -ls /  
Found 4 items  
drwxr-xr-x - dataflair supergroup 0 2020-01-29 14:42 /dataflair  
-rw-r--r-- 1 dataflair supergroup 38 2020-01-30 09:51 /filefromlocal  
drwxr-xr-x - dataflair supergroup 0 2020-01-29 11:30 /newDataFlair  
--w----- 1 dataflair supergroup 0 2020-01-29 17:17 /testfile  
dataflair@admin1-All-Series:~$
```

Hadoop HDFS put Command Description:

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.

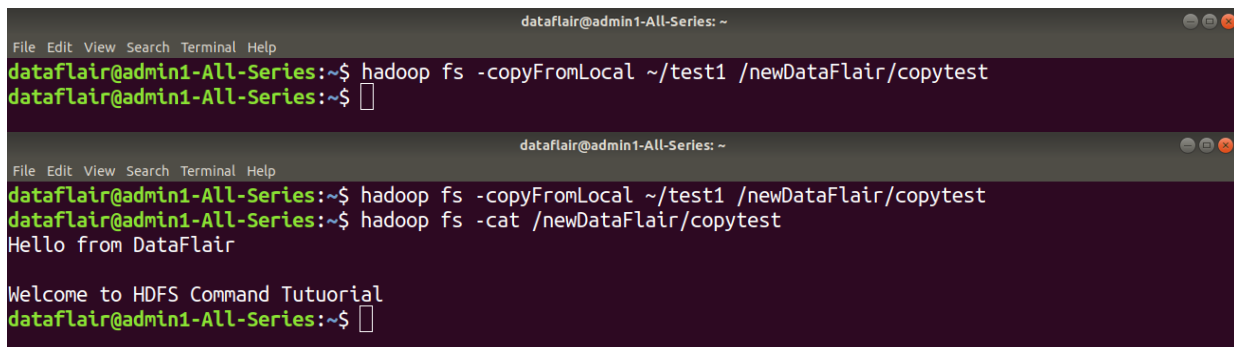
5. copyFromLocal

Hadoop HDFS copyFromLocal Command Usage:

`hadoop fs -copyFromLocal <localsrc> <hdfs destination>`

Hadoop HDFS copyFromLocal Command Example:

Here in the below example, we are trying to copy the 'test1' file present in the local file system to the newDataFlair directory of Hadoop.



```
dataflair@admin1-All-Series: ~  
File Edit View Search Terminal Help  
dataflair@admin1-All-Series:~$ hadoop fs -copyFromLocal ~/test1 /newDataFlair/copytest  
dataflair@admin1-All-Series:~$  
  
dataflair@admin1-All-Series: ~  
File Edit View Search Terminal Help  
dataflair@admin1-All-Series:~$ hadoop fs -copyFromLocal ~/test1 /newDataFlair/copytest  
dataflair@admin1-All-Series:~$ hadoop fs -cat /newDataFlair/copytest  
Hello from DataFlair  
  
Welcome to HDFS Command Tutorial  
dataflair@admin1-All-Series:~$
```

Hadoop HDFS copyFromLocal Command Description:

This command copies the file from the local file system to HDFS.

IoT & Applications (18EI2T09)

6. get

Hadoop HDFS get Command Usage:

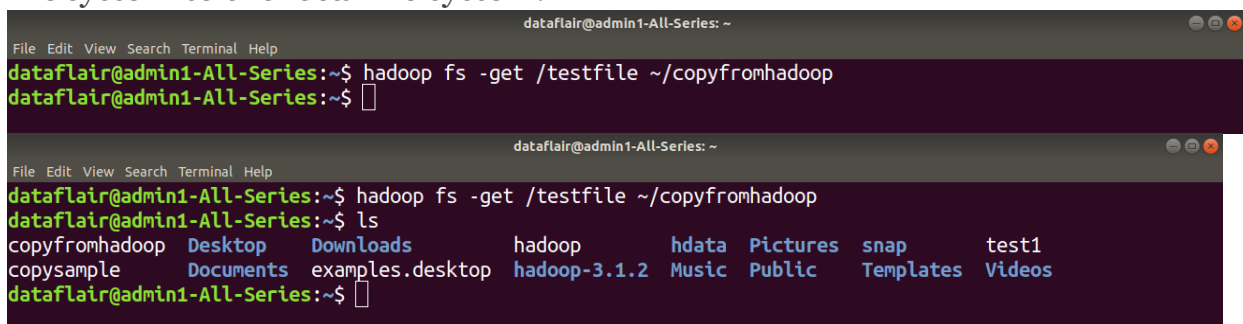
`hadoop fs -get <src> <localdest>`

Hadoop HDFS get Command Example:

In this example, we are trying to copy the 'testfile' of the hadoop filesystem to the local file system.

Hadoop HDFS get Command Description:

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



```
dataflair@admin1-All-Series: ~  
File Edit View Search Terminal Help  
dataflair@admin1-All-Series:~$ hadoop fs -get /testfile ~/copyfromhadoop  
dataflair@admin1-All-Series:~$  
  
dataflair@admin1-All-Series: ~  
File Edit View Search Terminal Help  
dataflair@admin1-All-Series:~$ hadoop fs -get /testfile ~/copyfromhadoop  
dataflair@admin1-All-Series:~$ ls  
copyfromhadoop Desktop Downloads hadoop hdata Pictures snap test1  
copysample Documents examples.desktop hadoop-3.1.2 Music Public Templates Videos  
dataflair@admin1-All-Series:~$
```

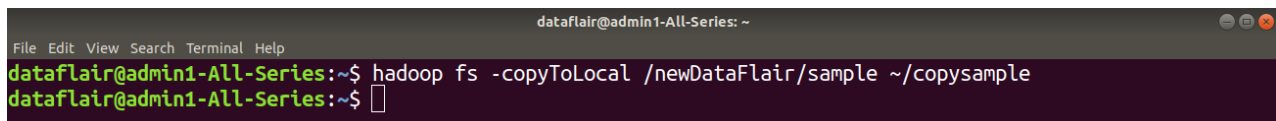
7. copyToLocal

Hadoop HDFS copyToLocal Command Usage:

`hadoop fs -copyToLocal <hdfs source> <localdst>`

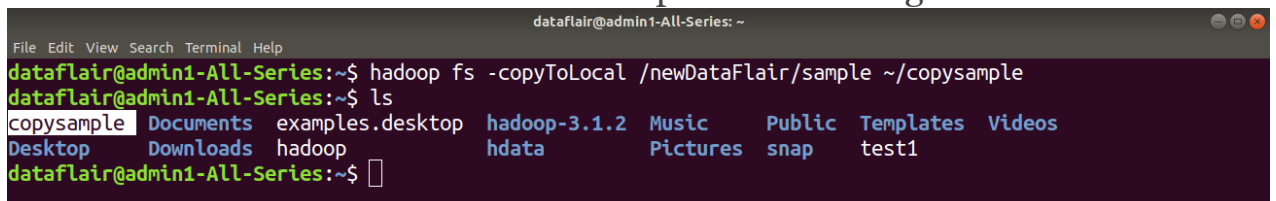
Hadoop HDFS copyToLocal Command Example:

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



```
dataflair@admin1-All-Series: ~  
File Edit View Search Terminal Help  
dataflair@admin1-All-Series:~$ hadoop fs -copyToLocal /newDataFlair/sample ~/copysample  
dataflair@admin1-All-Series:~$
```

We can cross-check whether the file is copied or not using the ls command.



```
dataflair@admin1-All-Series: ~  
File Edit View Search Terminal Help  
dataflair@admin1-All-Series:~$ hadoop fs -copyToLocal /newDataFlair/sample ~/copysample  
dataflair@admin1-All-Series:~$ ls  
copysample Documents examples.desktop hadoop-3.1.2 Music Public Templates Videos  
Desktop Downloads hadoop hdata Pictures snap test1  
dataflair@admin1-All-Series:~$
```

Hadoop HDFS copyToLocal Description:

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

IoT & Applications (18EI2T09)

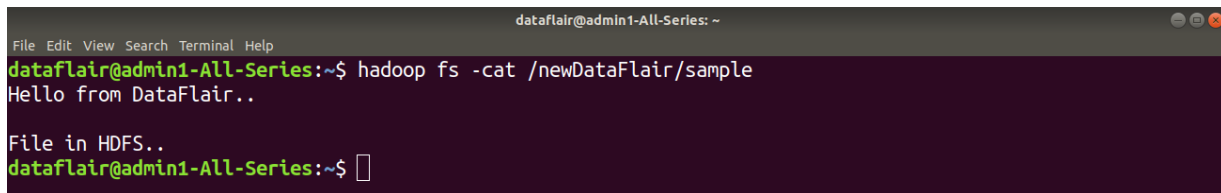
8. cat

Hadoop HDFS cat Command Usage:

`hadoop fs -cat /path_to_file_in_hdfs`

Hadoop HDFS cat Command Example:

Here in this example, we are using the cat command to display the content of the 'sample' file present in newDataFlair directory of HDFS.



```
dataflair@admin1-All-Series: ~  
File Edit View Search Terminal Help  
dataflair@admin1-All-Series:~$ hadoop fs -cat /newDataFlair/sample  
Hello from DataFlair..  
  
File in HDFS..  
dataflair@admin1-All-Series:~$
```

Hadoop HDFS cat Command Description:

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

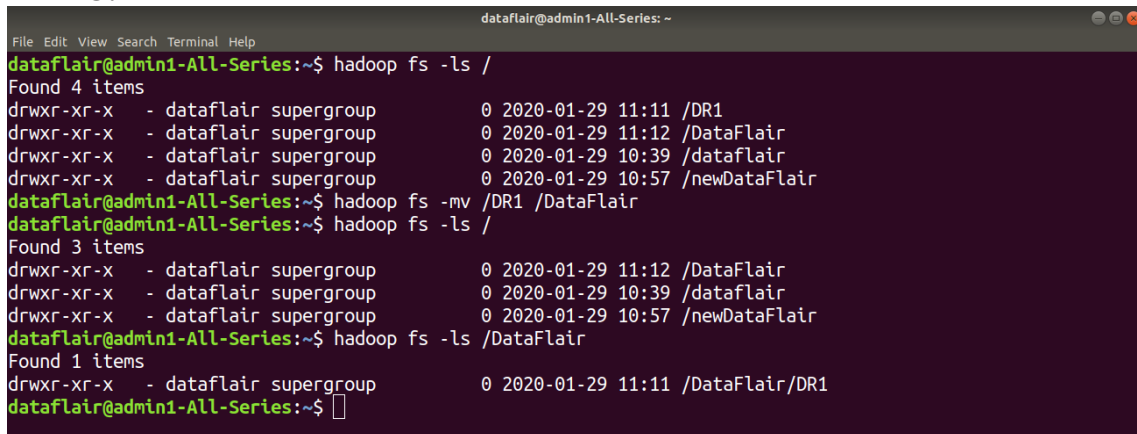
9. mv

Hadoop HDFS mv Command Usage:

`hadoop fs -mv <src> <dest>`

Hadoop HDFS mv Command Example:

In this example, we have a directory 'DR1' in HDFS. We are using mv command to move the DR1 directory to the DataFlair directory in HDFS.



```
dataflair@admin1-All-Series: ~  
File Edit View Search Terminal Help  
dataflair@admin1-All-Series:~$ hadoop fs -ls /  
Found 4 items  
drwxr-xr-x - dataflair supergroup 0 2020-01-29 11:11 /DR1  
drwxr-xr-x - dataflair supergroup 0 2020-01-29 11:12 /DataFlair  
drwxr-xr-x - dataflair supergroup 0 2020-01-29 10:39 /dataflair  
drwxr-xr-x - dataflair supergroup 0 2020-01-29 10:57 /newDataFlair  
dataflair@admin1-All-Series:~$ hadoop fs -mv /DR1 /DataFlair  
dataflair@admin1-All-Series:~$ hadoop fs -ls /  
Found 3 items  
drwxr-xr-x - dataflair supergroup 0 2020-01-29 11:12 /DataFlair  
drwxr-xr-x - dataflair supergroup 0 2020-01-29 10:39 /dataflair  
drwxr-xr-x - dataflair supergroup 0 2020-01-29 10:57 /newDataFlair  
dataflair@admin1-All-Series:~$ hadoop fs -ls /DataFlair  
Found 1 items  
drwxr-xr-x - dataflair supergroup 0 2020-01-29 11:11 /DataFlair/DR1  
dataflair@admin1-All-Series:~$
```

Hadoop HDFS mv Command Description:

IoT & Applications (18EI2T09)

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

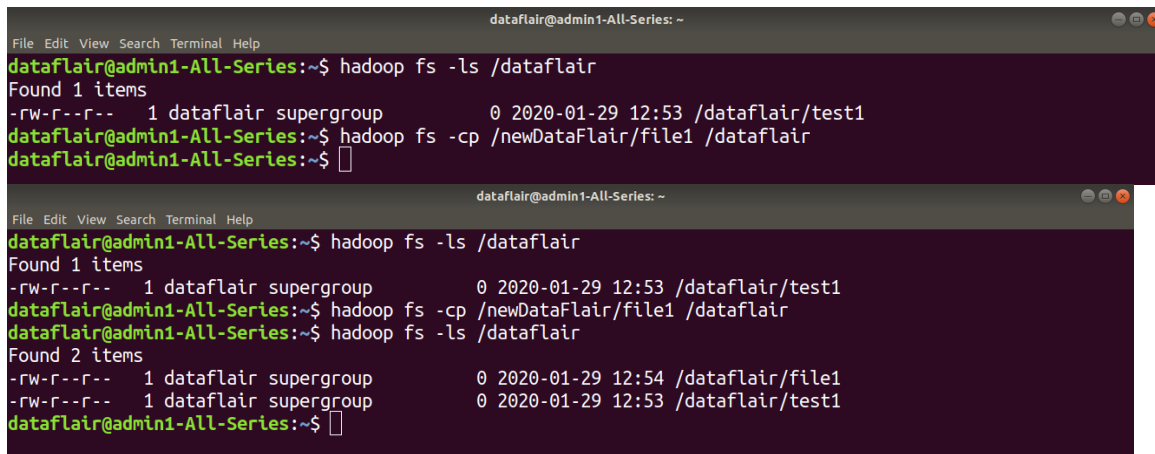
10. cp

Hadoop HDFS cp Command Usage:

```
hadoop fs -cp <src> <dest>
```

Hadoop HDFS cp Command Example:

In the below example we are copying the 'file1' present in newDataFlair directory in HDFS to the dataflair directory of HDFS.



```
dataflair@admin1-All-Series: ~  
File Edit View Search Terminal Help  
dataflair@admin1-All-Series:~$ hadoop fs -ls /dataflair  
Found 1 items  
-rw-r--r-- 1 dataflair supergroup 0 2020-01-29 12:53 /dataflair/test1  
dataflair@admin1-All-Series:~$ hadoop fs -cp /newDataFlair/file1 /dataflair  
dataflair@admin1-All-Series:~$  
  
dataflair@admin1-All-Series: ~  
File Edit View Search Terminal Help  
dataflair@admin1-All-Series:~$ hadoop fs -ls /dataflair  
Found 1 items  
-rw-r--r-- 1 dataflair supergroup 0 2020-01-29 12:53 /dataflair/test1  
dataflair@admin1-All-Series:~$ hadoop fs -cp /newDataFlair/file1 /dataflair  
dataflair@admin1-All-Series:~$ hadoop fs -ls /dataflair  
Found 2 items  
-rw-r--r-- 1 dataflair supergroup 0 2020-01-29 12:54 /dataflair/file1  
-rw-r--r-- 1 dataflair supergroup 0 2020-01-29 12:53 /dataflair/test1  
dataflair@admin1-All-Series:~$
```

Hadoop HDFS cp Command Description:

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

IoT & Applications (18EI2T09)

MapReduce Word Count Example

In MapReduce word count example, we find out the frequency of each word. Here, the role of Mapper is to map the keys to the existing values and the role of Reducer is to aggregate the keys of common values. So, everything is represented in the form of Key-value pair.

Pre-requisite

- **Java Installation** - Check whether the Java is installed or not using the following command.
`java -version`
- **Hadoop Installation** - Check whether the Hadoop is installed or not using the following command.
`hadoop version`

If any of them is not installed in your system, follow the below link to install it.

Steps to execute MapReduce word count example

- Create a text file in your local machine and write some text into it.
`$ nano data.txt`
- Check the text written in the data.txt file.
`$ cat data.txt`

In this example, we find out the frequency of each word exists in this text file.

- Create a directory in HDFS, where to kept text file.
`$ hdfs dfs -mkdir /test`
- Upload the data.txt file on HDFS in the specific directory.
`$ hdfs dfs -put /home/codegyani/data.txt /test`

Write the MapReduce program using eclipse.

IoT & Applications (18EI2T09)

File: WC_Mapper.java

```
1. package com.javatpoint;
2.
3. import java.io.IOException;
4. import java.util.StringTokenizer;
5. import org.apache.hadoop.io.IntWritable;
6. import org.apache.hadoop.io.LongWritable;
7. import org.apache.hadoop.io.Text;
8. import org.apache.hadoop.mapred.MapReduceBase;
9. import org.apache.hadoop.mapred.Mapper;
10. import org.apache.hadoop.mapred.OutputCollector;
11. import org.apache.hadoop.mapred.Reporter;
12. public class WC_Mapper extends MapReduceBase implements Mapper<LongWritable,Text,IntWritable>{
    ext,Text,IntWritable>{
13.     private final static IntWritable one = new IntWritable(1);
14.     private Text word = new Text();
15.     public void map(LongWritable key, Text value,OutputCollector<Text,IntWritable> output,
        tput,
16.         Reporter reporter) throws IOException{
17.         String line = value.toString();
18.         StringTokenizer tokenizer = new StringTokenizer(line);
19.         while (tokenizer.hasMoreTokens()){
20.             word.set(tokenizer.nextToken());
21.             output.collect(word, one);
22.         }
23.     }
24.
25. }
```

File: WC_Reducer.java

```
1. package com.javatpoint;
2.     import java.io.IOException;
3.     import java.util.Iterator;
```

IoT & Applications (18EI2T09)

```
4.    import org.apache.hadoop.io.IntWritable;
5.    import org.apache.hadoop.io.Text;
6.    import org.apache.hadoop.mapred.MapReduceBase;
7.    import org.apache.hadoop.mapred.OutputCollector;
8.    import org.apache.hadoop.mapred.Reducer;
9.    import org.apache.hadoop.mapred.Reporter;
10.
11.    public class WC_Reducer extends MapReduceBase implements Reducer<Text,IntWritable,Text,IntWritable> {
12.        public void reduce(Text key, Iterator<IntWritable> values,OutputCollector<Text,IntWritable> output,
13.            Reporter reporter) throws IOException {
14.            int sum=0;
15.            while (values.hasNext()) {
16.                sum+=values.next().get();
17.            }
18.            output.collect(key,new IntWritable(sum));
19.        }
20.    }
```

File: WC_Runner.java

```
1. package com.javatpoint;
2.
3.    import java.io.IOException;
4.    import org.apache.hadoop.fs.Path;
5.    import org.apache.hadoop.io.IntWritable;
6.    import org.apache.hadoop.io.Text;
7.    import org.apache.hadoop.mapred.FileInputFormat;
8.    import org.apache.hadoop.mapred.FileOutputFormat;
9.    import org.apache.hadoop.mapred.JobClient;
10.   import org.apache.hadoop.mapred.JobConf;
11.   import org.apache.hadoop.mapred.TextInputFormat;
12.   import org.apache.hadoop.mapred.TextOutputFormat;
13.   public class WC_Runner {
```

IoT & Applications (18EI2T09)

```
14.    public static void main(String[] args) throws IOException{
15.        JobConf conf = new JobConf(WC_Runner.class);
16.        conf.setJobName("WordCount");
17.        conf.setOutputKeyClass(Text.class);
18.        conf.setOutputValueClass(IntWritable.class);
19.        conf.setMapperClass(WC_Mapper.class);
20.        conf.setCombinerClass(WC_Reducer.class);
21.        conf.setReducerClass(WC_Reducer.class);
22.        conf.setInputFormat(TextInputFormat.class);
23.        conf.setOutputFormat(TextOutputFormat.class);
24.        FileInputFormat.setInputPaths(conf,new Path(args[0]));
25.        FileOutputFormat.setOutputPath(conf,new Path(args[1]));
26.        JobClient.runJob(conf);
27.    }
28. }
```

Download the source code.

- Create the jar file of this program and name it **countworddemo.jar**.
- Run the jar file
hadoop jar /home/codegyani/wordcountdemo.jar com.javatpoint.WC_Runner
/test/data.txt /r_output
- The output is stored in /r_output/part-00000
- Now execute the command to see the output.
hdfs dfs -cat /r_output/part-00000