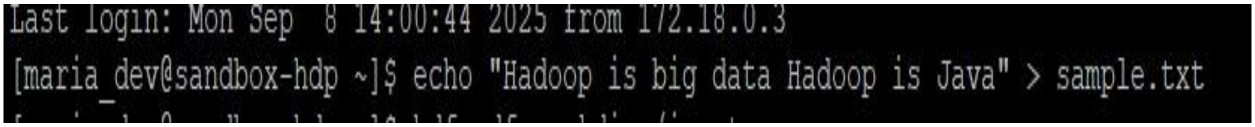
PRACTICAL-04

AIM:- Hadoop WordCount Execution Steps (Using PuTTY).

Step 1: Prepare Input Data

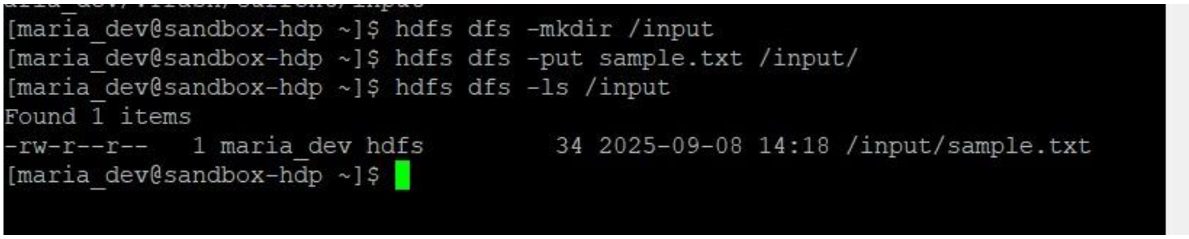
echo "Hadoop is big data Hadoop is Java" >sample.txt



```
Last login: Mon Sep  8 14:00:44 2025 from 172.18.0.3
[maria_dev@sandbox-hdp ~]$ echo "Hadoop is big data Hadoop is Java" > sample.txt
```

hdfs dfs -mkdir /input
hdfs dfs -put sample.txt /input/

hdfs dfs -ls /input



```
[maria_dev@sandbox-hdp ~]$ hdfs dfs -mkdir /input
[maria_dev@sandbox-hdp ~]$ hdfs dfs -put sample.txt /input/
[maria_dev@sandbox-hdp ~]$ hdfs dfs -ls /input
Found 1 items
-rw-r--r--   1 maria_dev hdfs         34 2025-09-08 14:18 /input/sample.txt
[maria_dev@sandbox-hdp ~]$
```

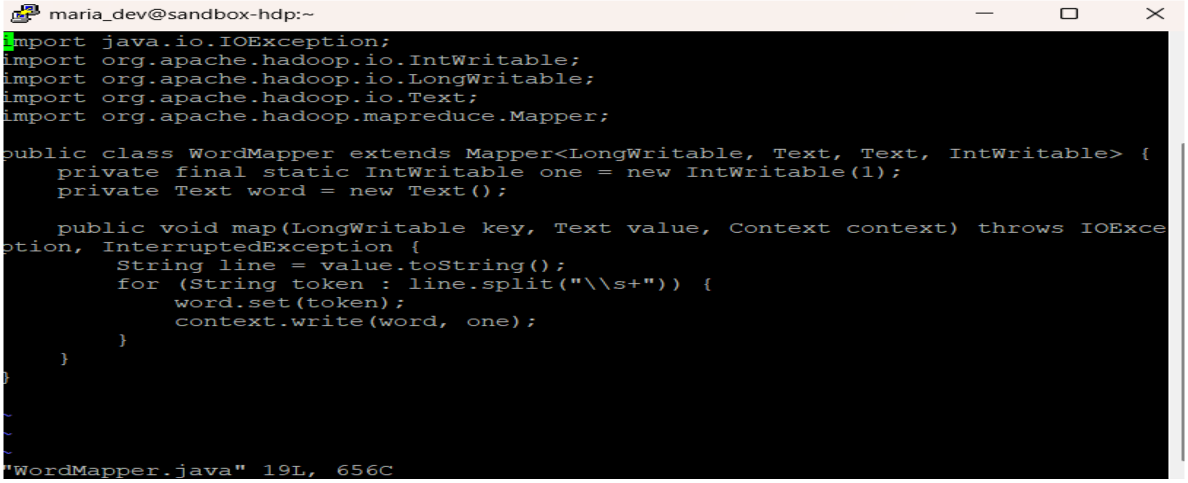
Step 2: Create Java Files

By using this code we will create the java files by using vi command and then pasting the above code and then save the file.

WordMapper.java

```
import java.io.IOException;
import
org.apache.hadoop.io.IntWritable;
import
org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
public class WordMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
   private final static IntWritable one = new IntWritable(1);
   private Text word = new Text();
   public void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException {
       String line = value.toString();
```

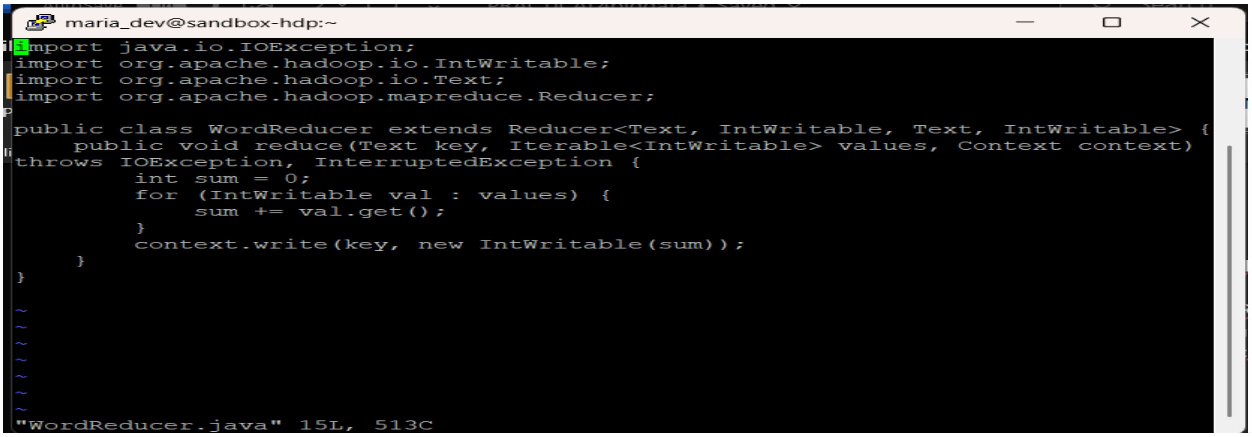
```
        for (String token : line.split("\\s+")) {  // Corrected escape sequence
            word.set(token);
            context.write(word, one);
        }

    }

}
```



```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class WordMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(LongWritable key, Text value, Context context) throws IOExce
ption, InterruptedException {
        String line = value.toString();
        for (String token : line.split("\\s+")) {
            word.set(token);
            context.write(word, one);
        }
    }
}
~
~
~
"WordMapper.java" 19L, 656C
```

WordReducer.java

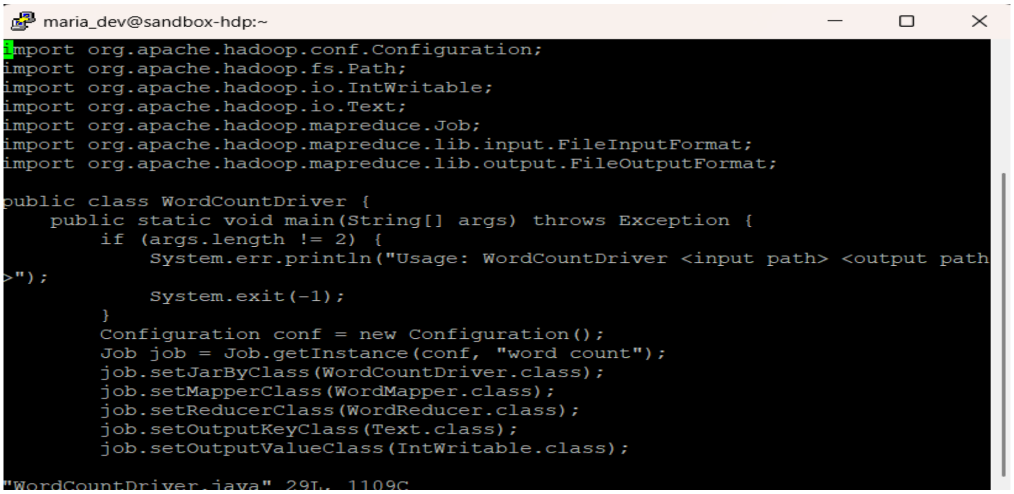
```
import java.io.IOException;
import
org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
public class WordReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
    public void reduce(Text key, Iterable<IntWritable> values, Context context) throws
IOException, InterruptedException {
        int sum = 0;
        for (IntWritable val :
            values) { sum +=
            val.get();
        }

        context.write(key, new IntWritable(sum));
    }

}
```



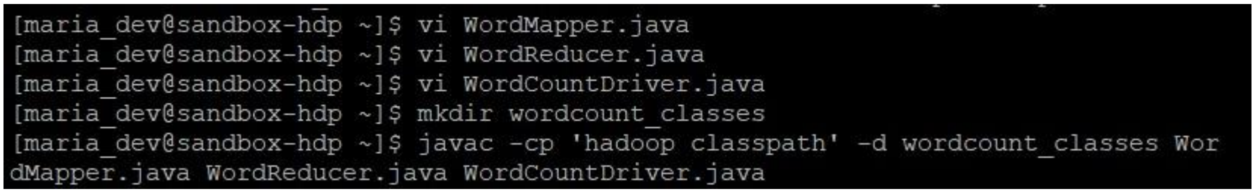
```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class WordReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
    public void reduce(Text key, Iterable<IntWritable> values, Context context)
throws IOException, InterruptedException {
        int sum = 0;
        for (IntWritable val : values) {
            sum += val.get();
        }
        context.write(key, new IntWritable(sum));
    }
}
~
~
~
~
~
~
~
"WordReducer.java" 15L, 513C
```

WordCountDriver.java

```
Importorg.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
public class WordCountDriver {
    public static void main(String[] args) throws Exception {
        if (args.length != 2) {
            System.err.println("Usage: WordCountDriver <input path> <output path>");
            System.exit(-1);
        }

        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "word count");
        job.setJarByClass(WordCountDriver.class);
        job.setMapperClass(WordMapper.class);
        job.setReducerClass(WordReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }

}
```



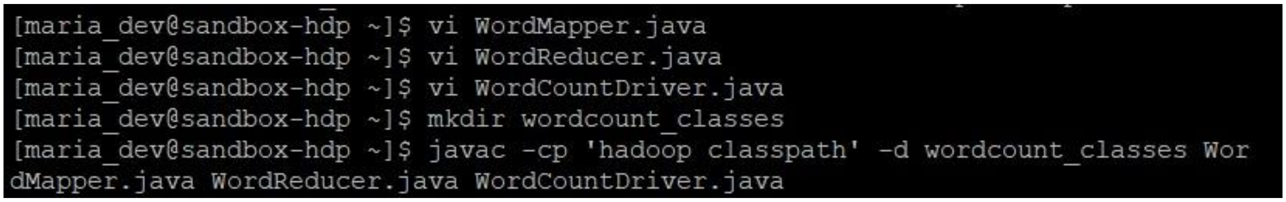
```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCountDriver {
    public static void main(String[] args) throws Exception {
        if (args.length != 2) {
            System.err.println("Usage: WordCountDriver <input path> <output path
>");
            System.exit(-1);
        }
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "word count");
        job.setJarByClass(WordCountDriver.class);
        job.setMapperClass(WordMapper.class);
        job.setReducerClass(WordReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

"WordCountDriver.java" 29L, 1109C
```



```
[maria_dev@sandbox-hdp ~]$ vi WordMapper.java
[maria_dev@sandbox-hdp ~]$ vi WordReducer.java
[maria_dev@sandbox-hdp ~]$ vi WordCountDriver.java
[maria_dev@sandbox-hdp ~]$ mkdir wordcount_classes
[maria_dev@sandbox-hdp ~]$ javac -cp 'hadoop classpath' -d wordcount_classes Wor
dMapper.java WordReducer.java WordCountDriver.java
```

Step 3: Compile Java Files

mkdir wordcount_classes



```
[maria_dev@sandbox-hdp ~]$ vi WordMapper.java
[maria_dev@sandbox-hdp ~]$ vi WordReducer.java
[maria_dev@sandbox-hdp ~]$ vi WordCountDriver.java
[maria_dev@sandbox-hdp ~]$ mkdir wordcount_classes
[maria_dev@sandbox-hdp ~]$ javac -cp 'hadoop classpath' -d wordcount_classes Wor
dMapper.java WordReducer.java WordCountDriver.java
```

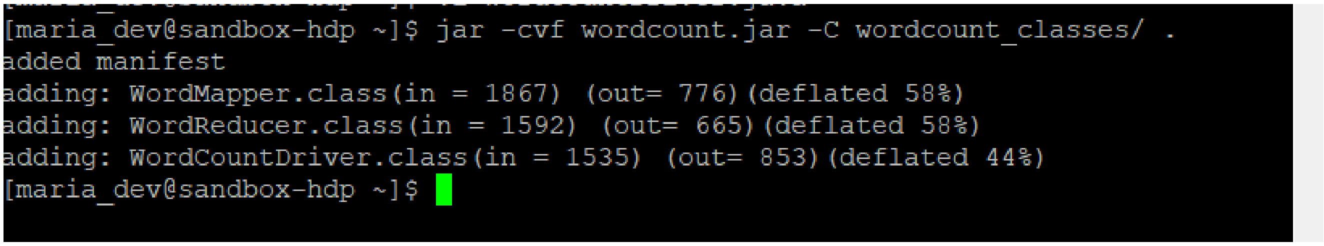
hadoop com.sun.tools.javac.Main -d wordcount_classes WordMapper.java WordReducer.java WordCountDriver.java

(If error, use:)

javac -cp `hadoop classpath` -d wordcount_classes WordMapper.java WordReducer.java WordCountDriver.java

Step 4: Create JAR

jar -cvf wordcount.jar -C wordcount_classes/ .

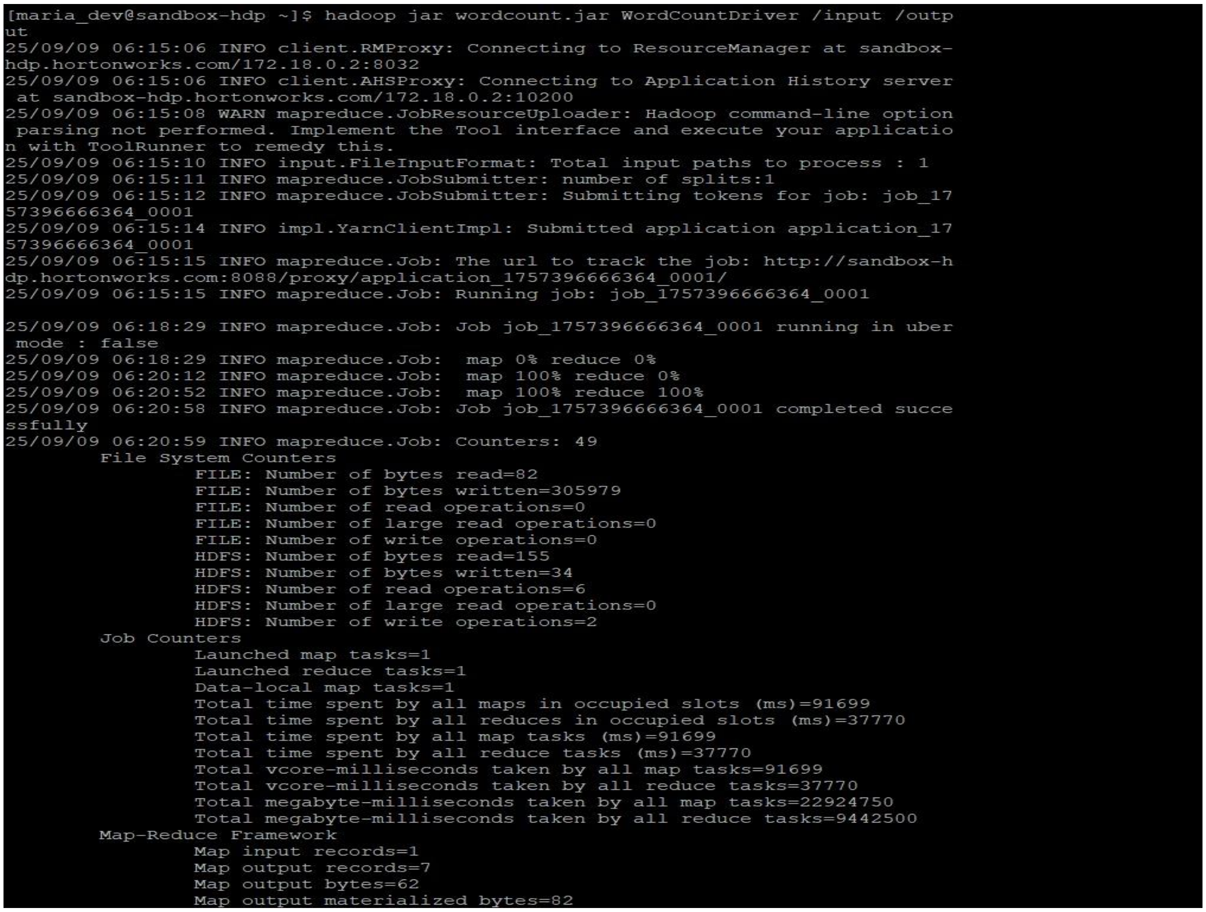


```
[maria_dev@sandbox-hdp ~]$ jar -cvf wordcount.jar -C wordcount_classes/ .
added manifest
adding: WordMapper.class(in = 1867) (out= 776)(deflated 58%)
adding: WordReducer.class(in = 1592) (out= 665)(deflated 58%)
adding: WordCountDriver.class(in = 1535) (out= 853)(deflated 44%)
[maria_dev@sandbox-hdp ~]$
```

Step 5: Run MapReduce Job

hdfs dfs -rm -r /output

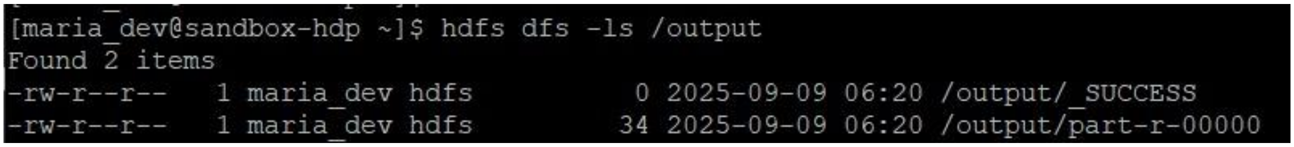
hadoop jar wordcount.jar WordCountDriver /input /output



```
[maria_dev@sandbox-hdp ~]$ hadoop jar wordcount.jar WordCountDriver /input /outp
ut
25/09/09 06:15:06 INFO client.RMProxy: Connecting to ResourceManager at sandbox-
hdp.hortonworks.com/172.18.0.2:8032
25/09/09 06:15:06 INFO client.AHSProxy: Connecting to Application History server
 at sandbox-hdp.hortonworks.com/172.18.0.2:10200
25/09/09 06:15:08 WARN mapreduce.JobResourceUploader: Hadoop command-line option
 parsing not performed. Implement the Tool interface and execute your applicatio
n with ToolRunner to remedy this.
25/09/09 06:15:10 INFO input.FileInputFormat: Total input paths to process : 1
25/09/09 06:15:11 INFO mapreduce.JobSubmitter: number of splits:1
25/09/09 06:15:12 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_17
57396666364_0001
25/09/09 06:15:14 INFO impl.YarnClientImpl: Submitted application application_17
57396666364_0001
25/09/09 06:15:15 INFO mapreduce.Job: The url to track the job: http://sandbox-h
dp.hortonworks.com:8088/proxy/application_1757396666364_0001/
25/09/09 06:15:15 INFO mapreduce.Job: Running job: job_1757396666364_0001

25/09/09 06:18:29 INFO mapreduce.Job: Job job_1757396666364_0001 running in uber
 mode : false
25/09/09 06:18:29 INFO mapreduce.Job:  map 0% reduce 0%
25/09/09 06:20:12 INFO mapreduce.Job:  map 100% reduce 0%
25/09/09 06:20:52 INFO mapreduce.Job:  map 100% reduce 100%
25/09/09 06:20:58 INFO mapreduce.Job: Job job_1757396666364_0001 completed succe
ssfully
25/09/09 06:20:59 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=82
                FILE: Number of bytes written=305979
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=155
                HDFS: Number of bytes written=34
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=91699
                Total time spent by all reduces in occupied slots (ms)=37770
                Total time spent by all map tasks (ms)=91699
                Total time spent by all reduce tasks (ms)=37770
                Total vcore-milliseconds taken by all map tasks=91699
                Total vcore-milliseconds taken by all reduce tasks=37770
                Total megabyte-milliseconds taken by all map tasks=22924750
                Total megabyte-milliseconds taken by all reduce tasks=9442500
        Map-Reduce Framework
                Map input records=1
                Map output records=7
                Map output bytes=62
                Map output materialized bytes=82
```

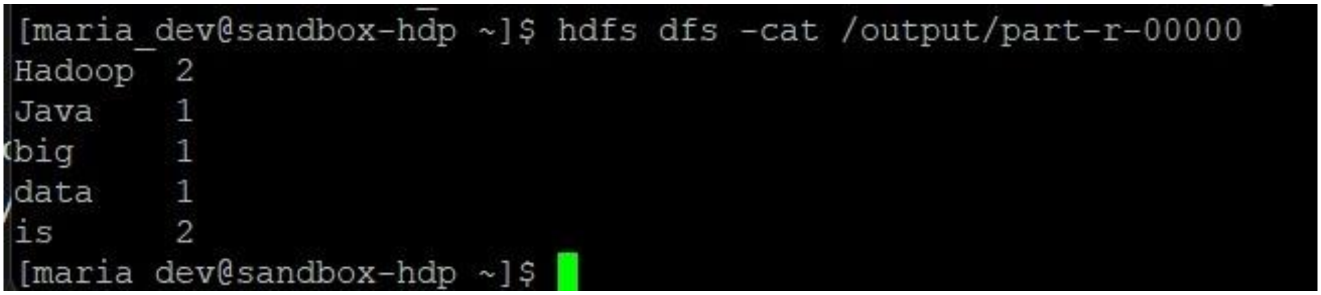
Step 6: View Output

hdfs dfs -ls /output



```
[maria_dev@sandbox-hdp ~]$ hdfs dfs -ls /output
Found 2 items
-rw-r--r--   1 maria_dev hdfs          0 2025-09-09 06:20 /output/_SUCCESS
-rw-r--r--   1 maria_dev hdfs         34 2025-09-09 06:20 /output/part-r-00000
```

hdfs dfs -cat /output/part-r-00000



```
[maria_dev@sandbox-hdp ~]$ hdfs dfs -cat /output/part-r-00000
Hadoop  2
Java    1
big     1
data    1
is      2
[maria dev@sandbox-hdp ~]$
```

CONCLUSION:- We have completed the Hadoop WordCount Exceution steps using Putty.