Follow the steps to perform Hadoop WordCount :

1. Open Terminal : **start-all.sh**

2. Inside terminal : **jps**
   Verify if you can see namenode, datanodes and resource managers after jps.

3. Open Eclipse Java IDE , and create a **New Java Project >** name it anything, I have given the name : doop.
   3.1 ***Make sure to select Java SE 1.8 version*** 👈
   3.2 ***No need to add any external JAR files.*** 🤝

4. You can now see **src** inside doop **,** created in it, select it and create **New Java Class** > name it **WordCount**. Similarly create 2 more classes, **WordMapper** and **IntSumReducer**

   NOTE : Delete the package name while creating these classes. Don't create any package with it. 👍

5. You can now see 3 .java files : WordCount.java , WordMapper.java and IntSumReducer.java inside the src folder.

6. Add the code in **WordCount.java :**

```
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 WordCount {
   public static void main(String[] args) throws Exception {
      Configuration conf = new Configuration();
      Job job = Job.getInstance(conf, "word count");
```

```
        job.setJarByClass(WordCount.class);
        job.setMapperClass(WordMapper.class);
        job.setCombinerClass(IntSumReducer.class);
        job.setReducerClass(IntSumReducer.class);

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

        FileInputFormat.addInputPath(job, new Path("input.txt"));
        FileOutputFormat.setOutputPath(job, new Path("output"));

        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

7. Similarly add the codes in **WordMapper.java :**

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

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

    public void map(Object key, Text value, Context context) throws    IOException,
InterruptedException {
        String[] tokens = value.toString().split("\\s+");
        for (String token : tokens) {
            word.set(token.toLowerCase().replaceAll("[^a-zA-Z]", ""));
            if (!word.toString().isEmpty()) {
                context.write(word, one);
            }
        }
    }
}
```

Add this in **IntSumReducer.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 IntSumReducer 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));
   }
}
```

8. Make sure you Project directory looks like this :

doop :
- **src :**
  - WordCount.java
  - WordMapper.java
  - IntSumReducer.java
- **bin** (& others maybe) .

9. Open this directory in the terminal. (make sure you are inside doop) :
   Create the directory **classes** in doop.

hadoop@sel-a1-216-06:~/Desktop/doop$ **mkdir -p classes**

Make sure you Project directory looks like this :

doop :
- **src** :
  - WordCount.java
  - WordMapper.java
  - IntSumReducer.java
- **classes** *//you can see this now*

bin (& other maybe) .

10. Now try running :

    hadoop@sel-a1-216-06:~/Desktop/doop$ **javac -classpath "$(hadoop classpath)" -d classes src/*.java**

    NOTE : if you get version errors try running: javac ***-source 1.8 -target 1.8*** -classpath "$(hadoop classpath)" -d classes src/*.java

11. Now your Project dir looks like :

doop :
- **src :**
  - WordCount.java
  - WordMapper.java
  - IntSumReducer.java
- **classes :**
  - WordCount.class
  - WordMapper.class
  - IntSumReducer.class
- **bin** (& other maybe) .

12. Now go inside classes from terminal and create the jar file of them :
    Open terminal :
    hadoop@sel-a1-216-06:~/Desktop/doop$ **cd classes**

    hadoop@sel-a1-216-06:~/Desktop/doop/classes$ **jar -cvf wordcount.jar *.class**

    Now your Project directory must look like :

doop :
- **src :**
  - WordCount.java
  - WordMapper.java
  - IntSumReducer.java
- **classes :**
  - WordCount.class
  - WordMapper.class
  - IntSumReducer.class

wordcount.jar *// this jar file is created*

bin (& other maybe) .

13. Again get back to doop in terminal :
    hadoop@sel-a1-216-06:~/Desktop/doop/classes$ **cd ..**

14. Go to Desktop, and inside doop create an input.txt file and add some text in it.

    Now your Project directory must look like :

doop :

src :

WordCount.java
WordMapper.java
IntSumReducer.java

classes :

WordCount.class
WordMapper.class
IntSumReducer.class
wordcount.jar

input.txt // this input file is created

bin (& other maybe)

15. Make a directory in hadoop filesystem and put the input.txt into it using the commands in terminal : (Make sure you are in doop in terminal)

    hadoop@sel-a1-216-06:~/Desktop/doop$ **hadoop fs -mkdir -p /user/hadoop**

    hadoop@sel-a1-216-06:~/Desktop/doop$ **hadoop fs -put input.txt /user/hadoop**

16. *To check if input is actually inside /user/hadoop too : you can do*
    hadoop@sel-a1-216-06:~/Desktop/doop$ ***hadoop fs -ls /user/hadoop***

    You can now see input.txt there too!

17. Now run this command from being in doop loc in terminal :
    hadoop@sel-a1-216-06:~/Desktop/doop$ **hadoop jar classes/wordcount.jar WordCount input.txt output**

18. The output is now created , to check output , use -cat command :
    hadoop@sel-a1-216-06:~/Desktop/doop$ **hadoop fs -cat /user/hadoop/output/part-r-00000**

And all done!
All the best!
🏆🙌

NOTE :

To remove input.txt and output from hadoop filesystem and to run it again and again, you can use -rm -r command :

For input.txt :
hadoop@sel-a1-216-06:~/Desktop/doop$ **hadoop fs -rm -r /user/hadoop/input.txt**

For output :
hadoop@sel-a1-216-06:~/Desktop/doop$ **hadoop fs -rm -r /user/hadoop/output**

~ by Siddhesh Jadhav - 31327