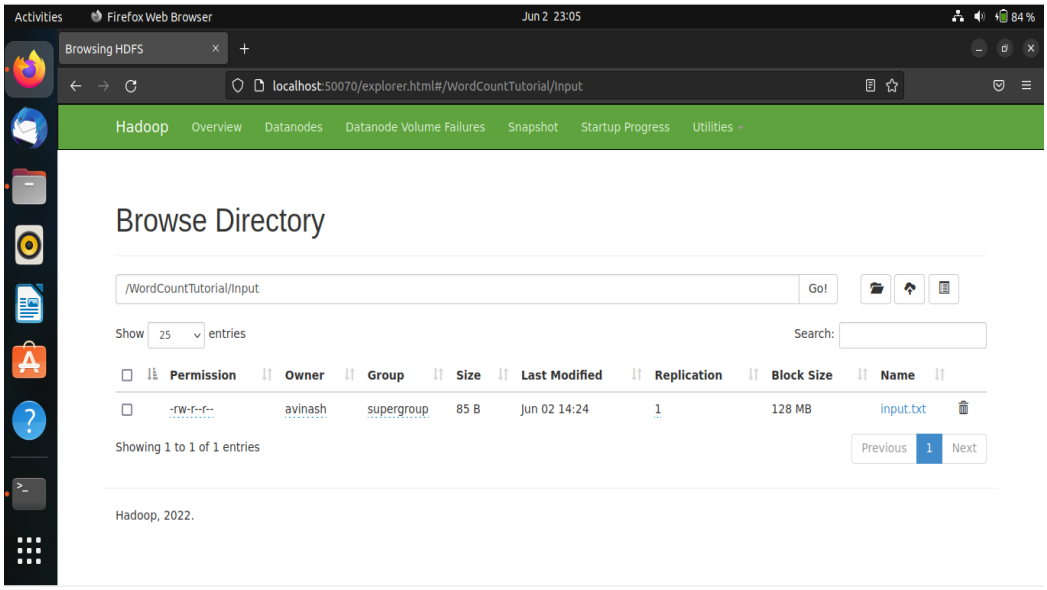
EX NO:13 Implementation of Map Reduce Program in Hadoop

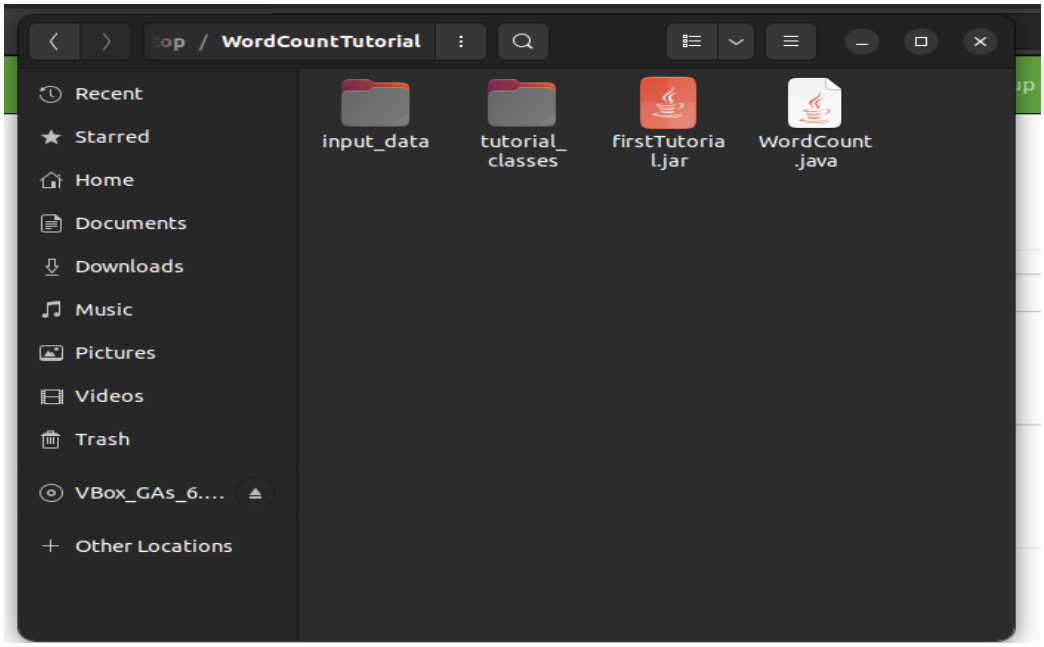
Aim:

To study the implementation of MapReduce Program in Hadoop.

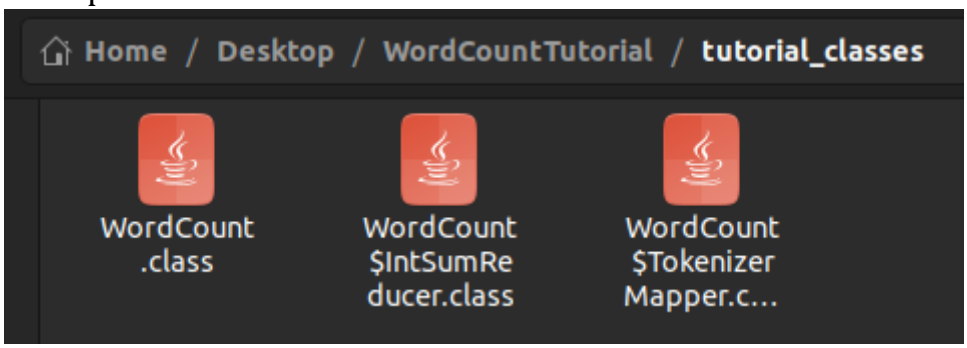
Procedure:

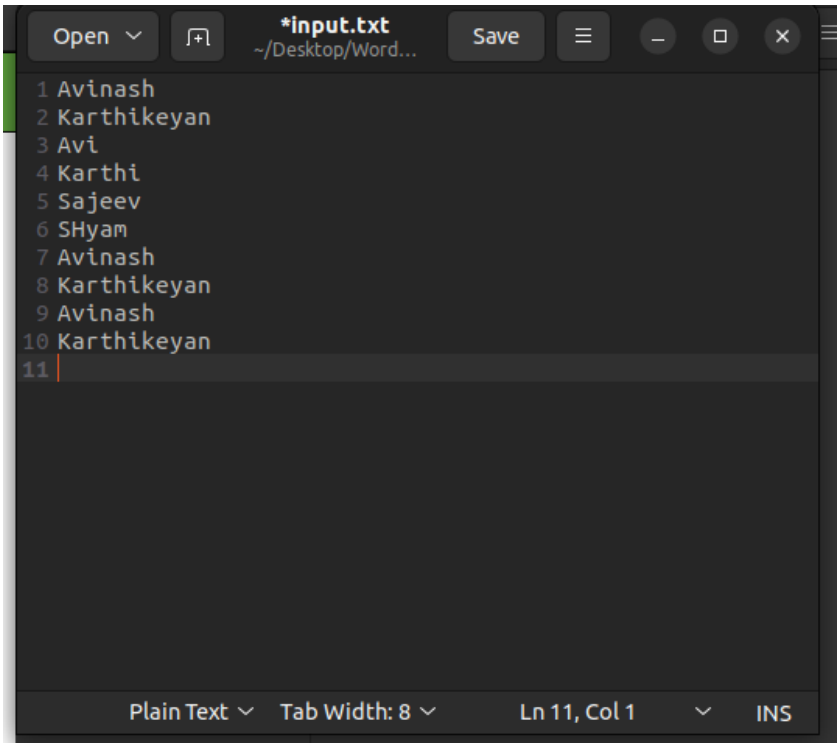
1. Install Hadoop in Ubuntu
2. Post installation of Hadoop in Ubuntu, the Hadoop will be running in the same machine, and when we navigate to localhost:50070, the filesystem will be available which we have created already





3. Let us create a folder named wordcount tutorial and have a input data file, and the same will contain the list of words which is to be fed as input to our Hadoop dfs





4. Wordcount.java is the file containing the source code for the core logic of map reduce

WordCount.java

```
import java.io.IOException;

import java.util.StringTokenizer;

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.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
public class WordCount {
 public static class TokenizerMapper
      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 {
    StringTokenizer itr = new StringTokenizer(value.toString());
    while (itr.hasMoreTokens()) {
      word.set(itr.nextToken());
      context.write(word, one);
    }

   }
 }
 public static class IntSumReducer
      extends Reducer<Text,IntWritable,Text,IntWritable> {
   private IntWritable result = new IntWritable();
   public void reduce(Text key, Iterable<IntWritable> values,
                      Context context
                      ) throws IOException, InterruptedException {
    int sum = 0;
    for (IntWritable val : values) {
      sum += val.get();
```

```
    }
    result.set(sum);
    context.write(key, result);
  }
 }

 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(TokenizerMapper.class);
  job.setCombinerClass(IntSumReducer.class);
  job.setReducerClass(IntSumReducer.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);
 }
}
```

5. After running the map reduce program with the input data specified , we will be getting the output .

```
avinash@Avinash:~/Desktop/WordCountTutorial$ hadoop jar '/home/avinash/Desktop/W
ordCountTutorial/firstTutorial.jar' WordCount /WordCountTutorial/Input /WordCoun
tTutorial/Output
22/06/02 23:02:29 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
22/06/02 23:02:30 INFO Configuration.deprecation: session.id is deprecated. Inst
ead, use dfs.metrics.session-id
22/06/02 23:02:30 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName
=JobTracker, sessionId=
Exception in thread "main" org.apache.hadoop.mapred.FileAlreadyExistsException:
Output directory hdfs://localhost:54310/WordCountTutorial/Output already exists
        at org.apache.hadoop.mapreduce.lib.output.FileOutputFormat.checkOutputSp
ecs(FileOutputFormat.java:146)
        at org.apache.hadoop.mapreduce.JobSubmitter.checkSpecs(JobSubmitter.java
:279)
        at org.apache.hadoop.mapreduce.JobSubmitter.submitJobInternal(JobSubmitt
er.java:145)
        at org.apache.hadoop.mapreduce.Job$11.run(Job.java:1573)
        at org.apache.hadoop.mapreduce.Job$11.run(Job.java:1570)
        at java.security.AccessController.doPrivileged(Native Method)
        at javax.security.auth.Subject.doAs(Subject.java:422)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInforma
tion.java:1938)
        at org.apache.hadoop.mapreduce.Job.submit(Job.java:1570)
```

```
avinash@Avinash:~/Desktop/WordCountTutorial$ hadoop dfs -cat /WordCountTutorial/
Output/*
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.

22/06/02 23:02:37 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
cat: `/WordCountTutorial/Output/*': Is a directory
Avi     1
Avinash 3
Karthi  1
Karthikeyan     3
SHyam   1
Sajeev  1
```

RESULTS:

Thus, the implementation of Map Reduce in Hadoop has been studied

REFERENCES:

https://www.youtube.com/watch?v=6sK3LDY7Pp4

https://www.youtube.com/watch?v=CtOhsZ0Sb1E