DATE: 16-6-2022

NAME: SUCHITHRA RAJ M

USN: 1NT19IS162

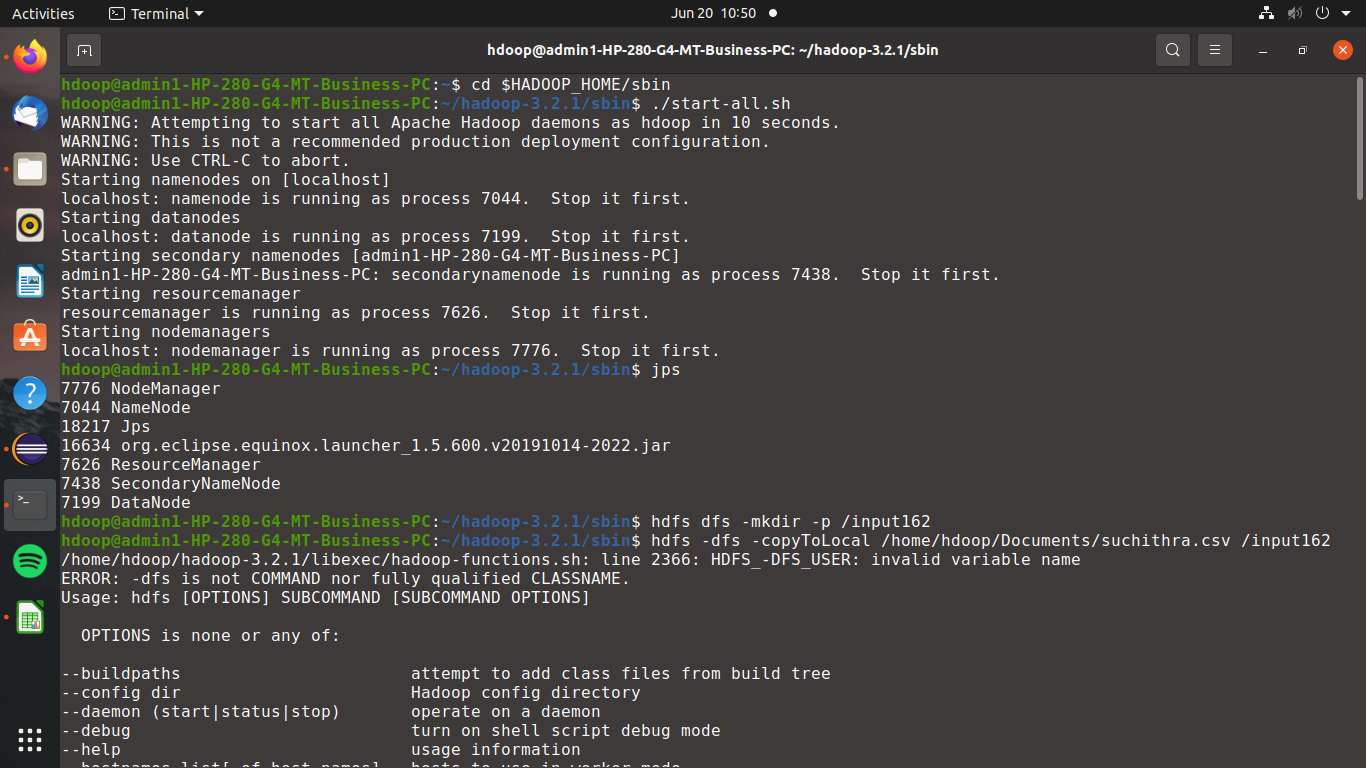
BD LAB

Use the Hadoop framework to write a MapReduce program to read a .csv file into a single node Hadoop cluster containing following fields Sl. No. CARD name UserName Amount withdrawn Implement the following, 1. Count the Number of transactions done by each user 2. Find the total amount of money transacted by each user

package suchithra162;  
  
import java.io.IOException;  
import java.util.\*;  
import org.apache.hadoop.fs.Path;  
import org.apache.hadoop.io.\*;  
import org.apache.hadoop.mapred.\*;  
  
  
public class TransactionCount {  
//MAPPER CODE  
  
public static class Map extends MapReduceBase implements  
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, OutputCollector<Text,  
IntWritable> output, Reporter reporter) throws IOException {  
String myString = value.toString();  
String[] userCount = myString.split(",");  
output.collect(new Text(userCount[3]), one);  
  
}  
}  
  
//REDUCER CODE  
public static class Reduce extends MapReduceBase implements  
Reducer<Text, IntWritable, Text, IntWritable> {  
public void reduce(Text key, Iterator<IntWritable> values,  
OutputCollector<Text, IntWritable> output, Reporter reporter) throws  
IOException { //{little: {1,1}}  
int finaluserCount = 0 ;  
Text mykey = key ;  
while(values.hasNext()) {  
IntWritable value = values.next();  
finaluserCount += value.get();  
}  
output.collect(mykey, new IntWritable(finaluserCount));  
}  
}  
//DRIVER CODE  
public static void main(String[] args) throws Exception {  
JobConf conf = new JobConf(TransactionCount.class);  
conf.setJobName("wordcount");  
conf.setOutputKeyClass(Text.class);  
conf.setOutputValueClass(IntWritable.class);  
conf.setMapperClass(Map.class);  
conf.setCombinerClass(Reduce.class);  
conf.setReducerClass(Reduce.class);  
conf.setInputFormat(TextInputFormat.class);  
conf.setOutputFormat(TextOutputFormat.class);  
FileInputFormat.setInputPaths(conf, new Path(args[0]));  
FileOutputFormat.setOutputPath(conf, new Path(args[1]));  
JobClient.runJob(conf);  
}  
  
}

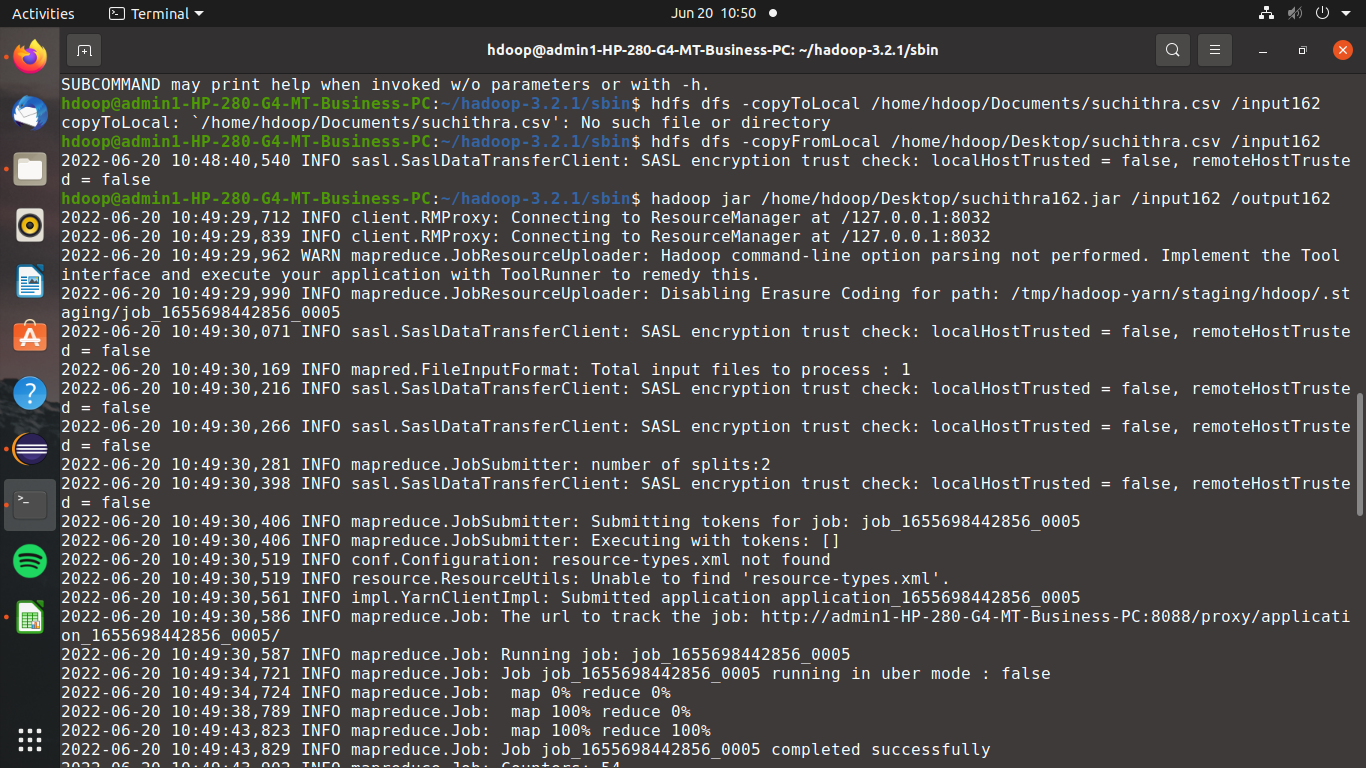
creating input directory.

Hdfs dfs -mkdir -p /input162



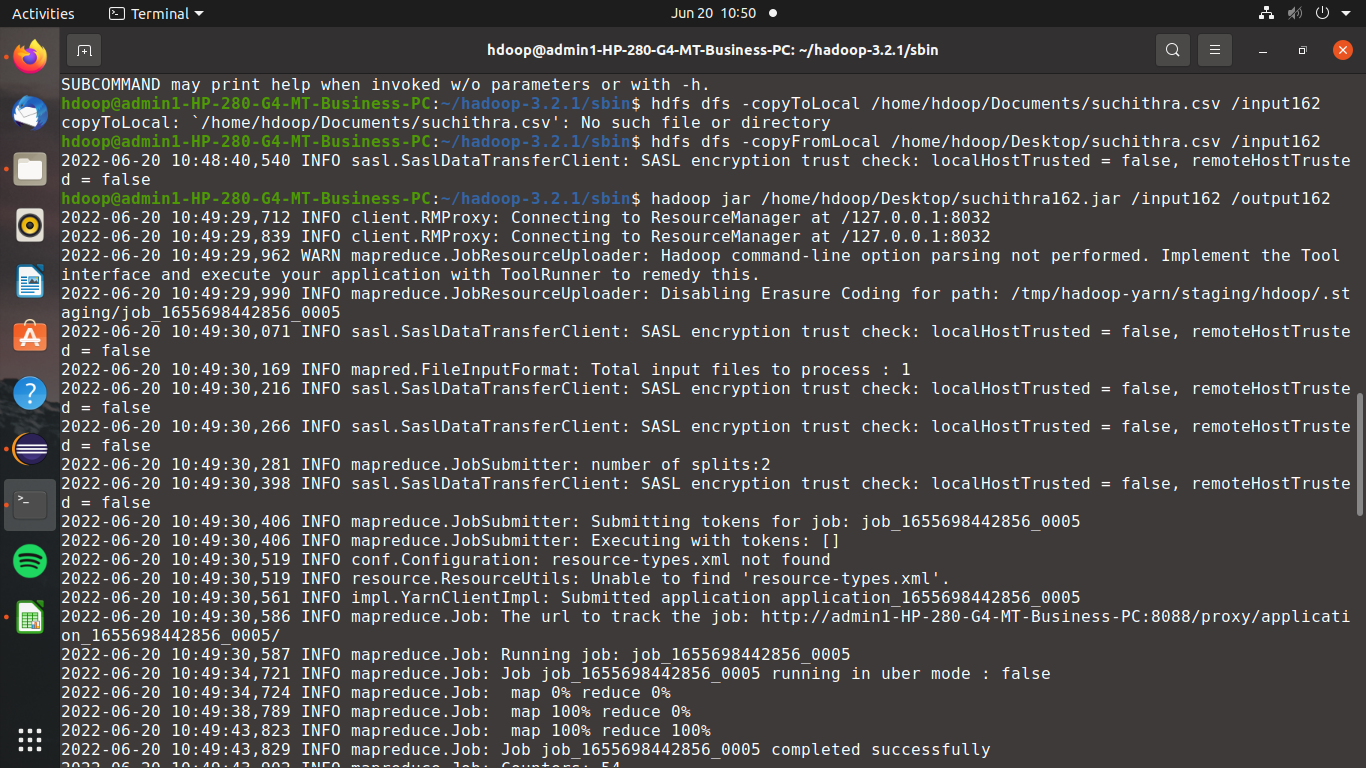
Coping the suchithra.csv file from local to hdfs.

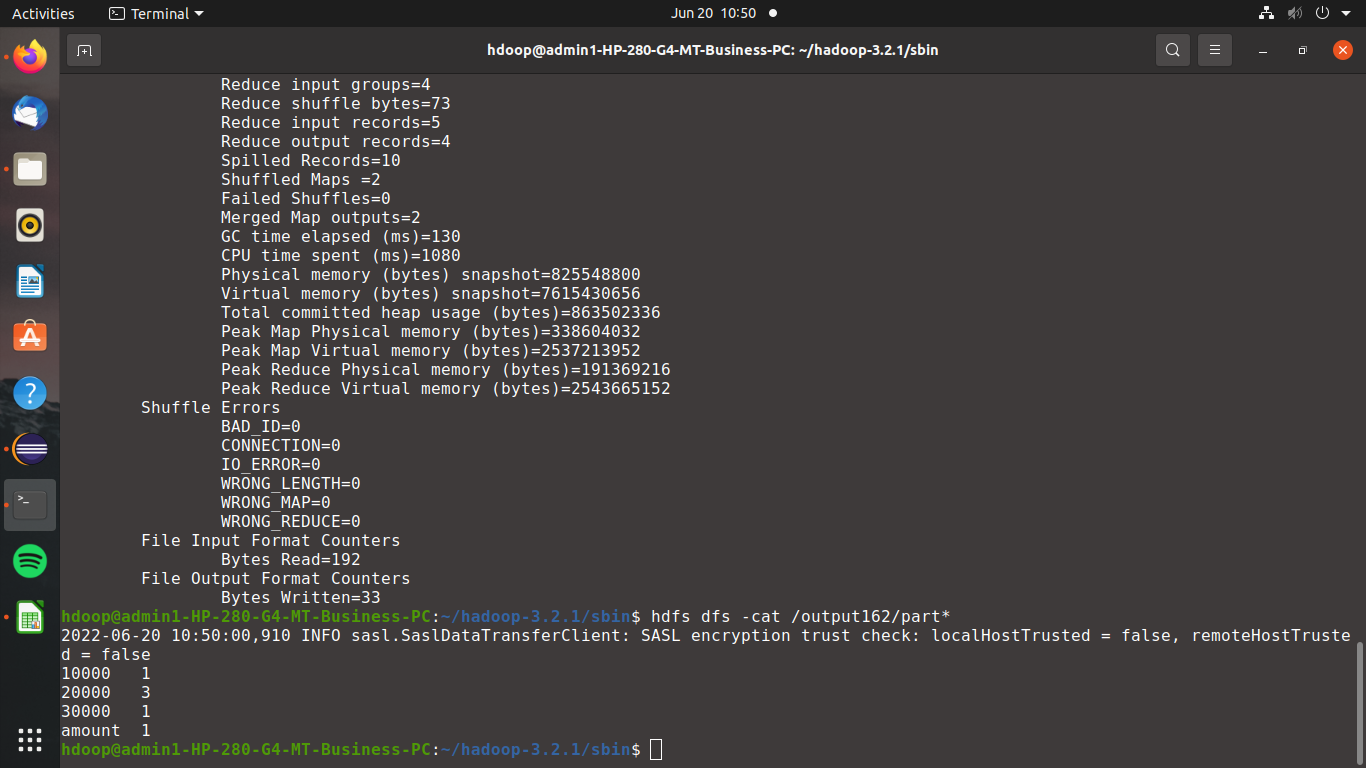
Hdfs dfs -copyFromLocal /home/hdoop/Desktop/suchithra.csv /input162



Hadoop jar /home/hdoop/Desktop/suchithra162.jar /input162 /output162

By using this above command, we can print the class path needed to get the Hadoop jar and required libraries.





For output we use hdfs dfs -cat /output162/part\*

