**Write a program in Map Reduce for WordCount operation.**

Step 1: Open virtual box and then start cloudera quickstart

Step 2: Open eclipse present on the cloudera desktop

Step 3: Create java project

File->New-> Java Project

Give project name: WordCount

Click Next

Step 4: Add Hadoop libraries to project

Select Libraries tab->click on Add External Jars

File System->user->lib->Hadoop

Select all library(jar) files->ok

Again click on Add Externa Jars

File System->user->lib->Hadoop->client

Select all library(jar) files from client->ok->finish

Step 5: Write java code for word count

Right click on src folder of project WordCount

New->class

Write class name WordCountDriver

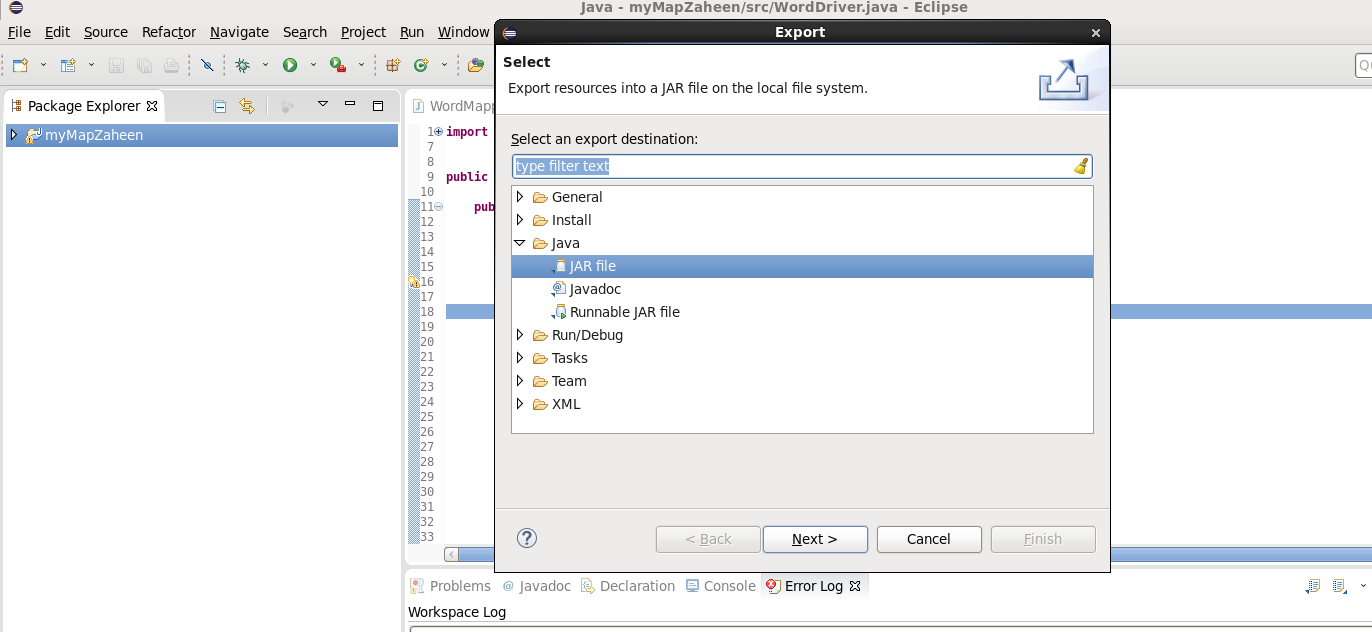
Click Finish

Write the code for WordCountDriver

Same way create classes WordMapper and SumReducer

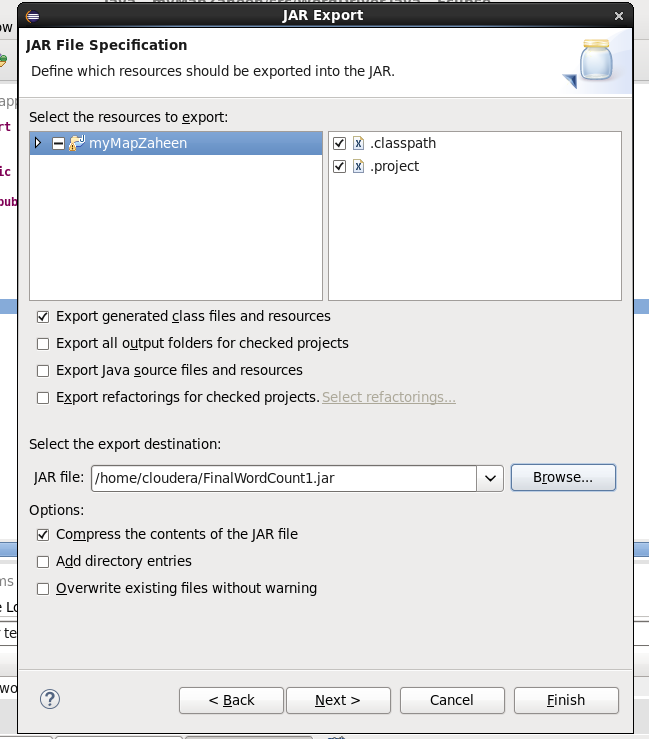
Step 6: Export the project as jar

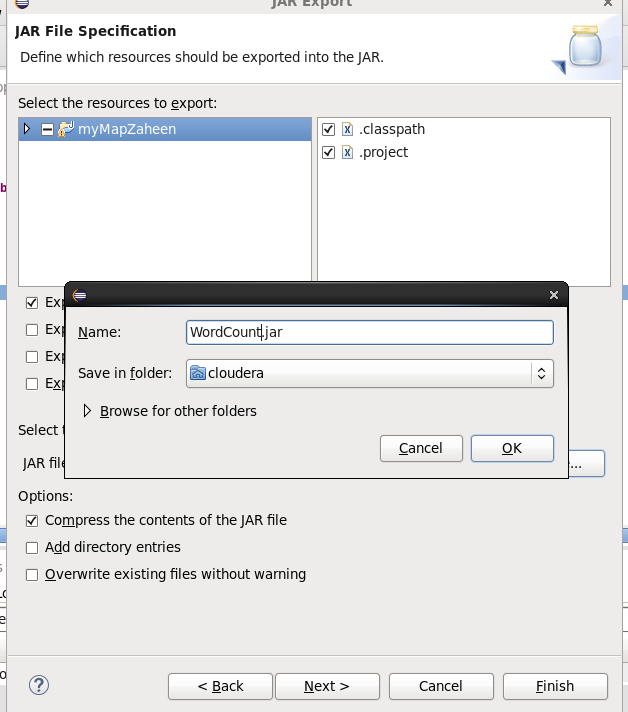
Right click on project WordCount and select Export>> Java>>JAR file>>Next



Select the export destination-Click browse-give file name wordcount.jar

Click ok>>Finish>>ok





Verify the jar file

Verify the jar file through command line open terminal give command

**ls**

step 7: Move the jar file to the Hadoop file system

**hdfs dfs -put wordcount.jar /user/cloudera**

**hdfs dfs -ls**

Step 8: Create the input file for the MapReduce program

Command: **cat > myInputFile.txt**

**Welcome to the NMITD MCA**

**I am persuing MCA in NMITD**

Enter data in input file and press enter and ctrl z

Command: **cat myInputFile.txt**

Step 9: Move the input file to the Hadoop file system

**hdfs dfs -put myInputFile.txt /user/cloudera**

**hdfs dfs -ls**

**hdfs dfs –ls /** (here / indicates root directory of Hadoop file system(hdfs))

step 10: Run mapreduce program on Hadoop

syntax: hadoop jar jarfilename.jar classname inputfilename.txt outputfoldername

command: **hadoop jar wordcount.jar WordCountDriver myInputFile.txt myOutput**

step 11: view output directory

**hdfs dfs –ls**

**hdfs dfs -ls /user/cloudera/myOutput**

step 12: view the output file

**hdfs dfs -cat /user/cloudera/myOutput/part-r-00000**

Code

**WordCountDriver.java**

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.mapreduce.Job;

public class WordCountDirver {

public static void main(String[] args) throws Exception {

if (args.length != 2) {

System.out.printf("Usage: WordCount <input dir> <output dir>\n");

System.exit(-1);

}

Job job = new Job();

job.setJarByClass(WordCountDriver.class);

job.setJobName("Word Count");

FileInputFormat.setInputPaths(job, new Path(args[0]));

FileOutputFormat.setOutputPath(job, new Path(args[1]));

job.setMapperClass(WordMapper.class);

job.setReducerClass(SumReducer.class);

job.setMapOutputKeyClass(Text.class);

job.setMapOutputValueClass(IntWritable.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

boolean success = job.waitForCompletion(true);

System.exit(success ? 0 : 1);

}

}

**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> {

@Override

public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {

String line = value.toString();

for (String word : line.split("\\W+")) {

if (word.length() > 0) {

context.write(new Text(word), new IntWritable(1));

}

}

}

}

**SumReducer.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 SumReducer extends Reducer<Text, IntWritable, Text, IntWritable> {

@Override

public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException {

int wordCount = 0;

for (IntWritable value : values) {

wordCount += value.get();

}

context.write(key, new IntWritable(wordCount));

}

}