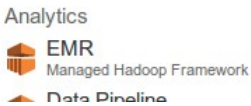
Word count example

Abdalrahman Alsaedi

To run word count in AWS you have two different ways; either use the already exist WordCount program, or to write your own file.

First: Using AWS word count program



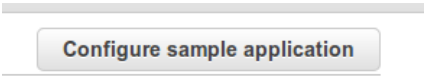
Prerequisites

a. S3 bucket
b. EC2 key pair

Steps

1. Login to your account in AWS
2. From **Amazon Web Services**, go to EMR
3. Click on Create Cluster
4. From Create Cluster page choose **Go to advance option**

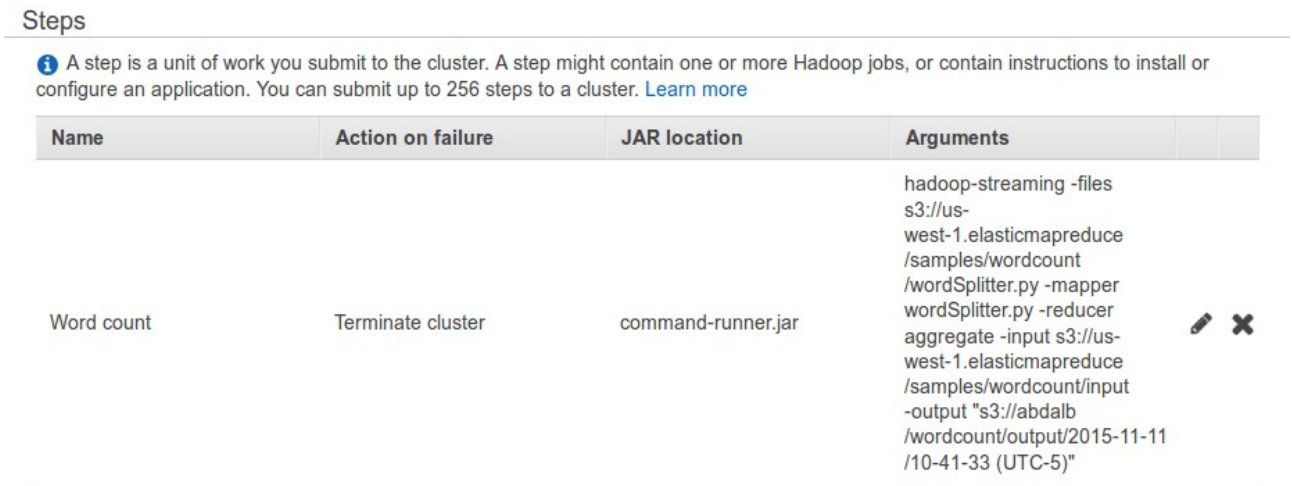
5. Now click on Configure sample application



6. From **Select sample application**, click the drop-down menu and select **Word Count**. In Output location just change the <*bucket-name*> with one of your buckets that found in S3. Click **Ok**

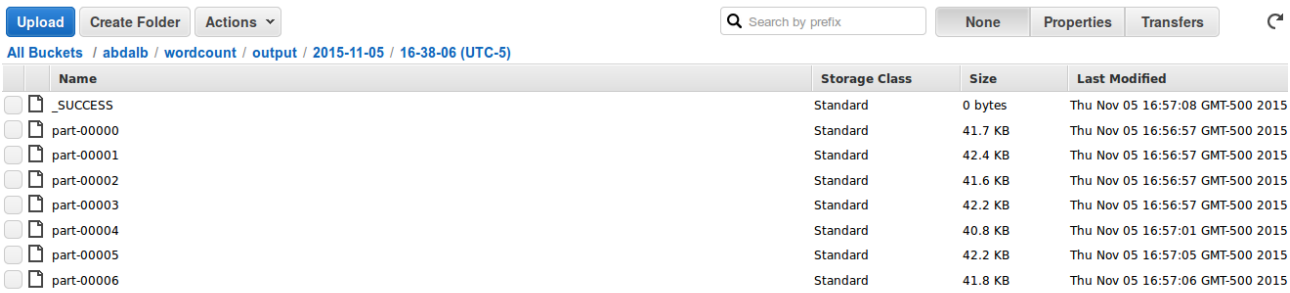
7. In Security and Access part choose your key pair from the drop-down menu

8. As a notification you will see in **Steps** part that you have one step called **Word count.**



9. Click Create Cluster. The page of creating cluster will appear and the state of your cluster is Starting. The cluster now in creating process and after the creation finished the state of your cluster will be changed to Running.

10. Go to S3 console and choose the bucket that you choose in step 6. you will find the output folder that contains the output, its some thing like in figure below.



Second: Create your own word count program using Eclipse

As a another way that we can create our own file of word count, we can use Eclipse to write the word count program, as following.

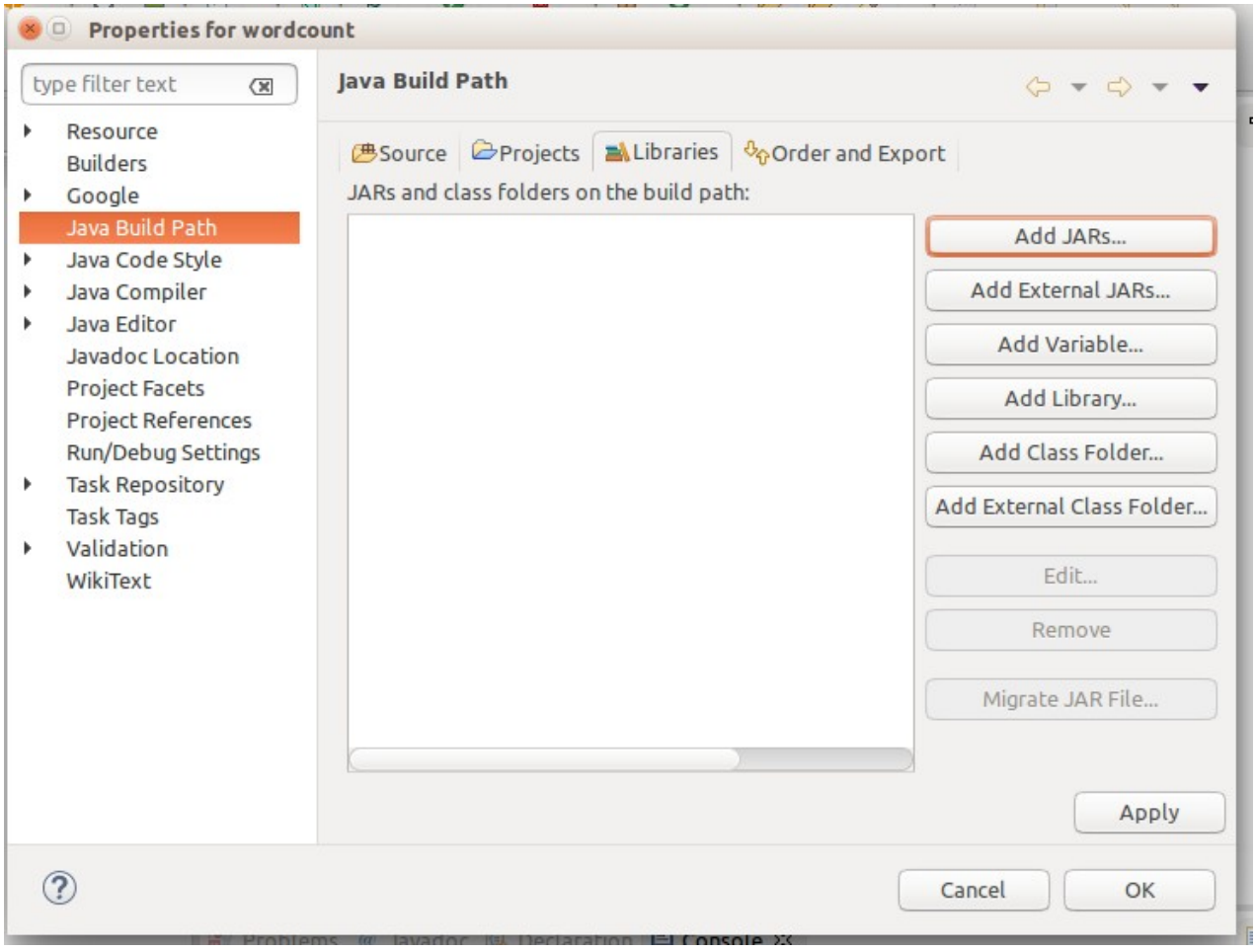
Prerequisites:

a. install Eclipse in your machine.
b. install Hadoop in your machine.

Note: you will find links to install above packages in **Topic Covered** of *Dr Gupta* web page

Steps

1. Start Eclipse
2. from file go to New project, then select Java project.
3. enter the name for your project, let say wordcount, and press finish.
4. Now we need to include some Hadoop dependenies to our project. To do so right click on the project name in project browser, and choose build bath and then Configure Build Path...



5. click on libraries , then Add external JARs then go to the location where you installed hadoop in your machine, select all JARs. Click Apply, and then OK.

6. We need to add more JARs. Do same thing that we did in step 5 but this time we will choose Client subfolder that found in Haddop folder. Select all JARs. Click Apply, and then OK.

7. Now our project redy to write some code in it. Right click on the project name and select new then package, name the package as wordcount, click finsh. Now right click on scr folder and select new and then class, name the class wordcount and click finish.

8. copy the code the following code and paste it in the class.

```
package wordcount;

import java.io.IOException;
import java.util.*;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapred.*;
import org.apache.hadoop.util.*;

public class WordCount {
  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 line = value.toString();
      StringTokenizer tokenizer = new StringTokenizer(line);
      while (tokenizer.hasMoreTokens()) {
        word.set(tokenizer.nextToken());
        output.collect(word, one);
      }
    }
  }
  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 {
      int sum = 0;
      while (values.hasNext()) {
        sum += values.next().get();
      }
      output.collect(key, new IntWritable(sum));
    }
  }
    public static void main(String[] args) throws Exception {
    JobConf conf = new JobConf(WordCount.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);
  }
}
```

9. Note here that in some cases you find tow errors that indicated in step ***public class WordCount {*** and in step ***JobConf conf = new JobConf(WordCount.class)***; this because your class note exactly has the same name, so change your class ofr change this name as your class.

10. Now we need to compile the class before export it. First click on the pakege name and go to Run As, then Run Configuration, in Main class type the name of your class, and click Apply, and then Run.

11. As a last step. Go to file>Export>Java>JAR file, then select your file and give it a name, and specific location. You need this location, because you will upload this JAR to S3 later.

12. To Run this file in AWS we follow steps from 1 to 4 that we mentioned in **First: Using AWS word count program.** Also we need to upload our JAR to one of the S3 buckets that we have, and we need to upload the file that we want to count the word on it.

Note Some times you will find that there is some other applications that will be installed in your cluster this will be mentioned in section Software Application/ Application to be installed, you do not need any application other than Hadoop, so delete all other application by pressing X.

13. In Security and Access part choose your key pair from the drop-down menu.

14. In step section select Custom JAR from the drop-down menu and press Configure and add . Give a name for your step, let say WordCountCS6030. In JAR location browse the location that you saved you JAR in it. In Argument we need to specify the location of the file that the word count will work on it and the location of the folder that will contains the output file(s), write the following:

s3n://<location_of_text_file>/Your_text_file s3n://<output_bucket>/

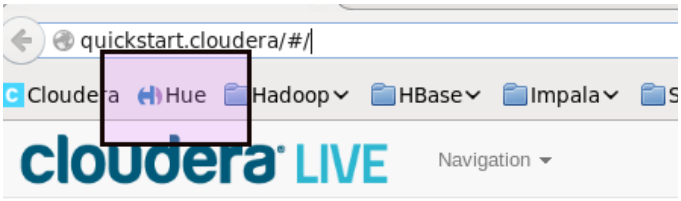
15. in Action on failure you can specify what AWS will do if your cluster fail to run your JAR. Specify the action that you want, let say Terminate cluste. Finally click Add.

16. Click Create Cluster. The page of creating cluster will appear and the state of your cluster is Starting. The cluster now in creating process and after the creation finished the state of your cluster will be changed to Running.

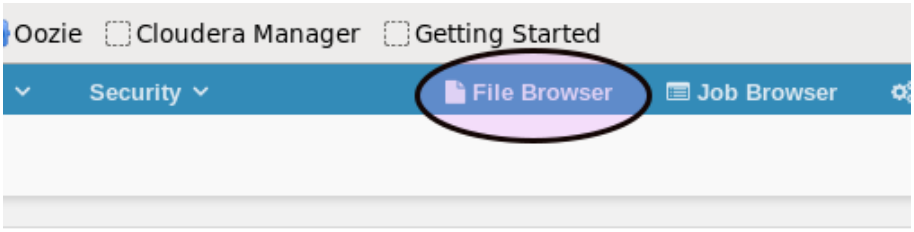
17. Go to S3 console and choose the bucket that you choose as an output bucket. you will find the output folder that contains the output.

Third: Using Hue to run word count program

To run the word count example in Hue follow the following steps:
1. Run your Cloudera VM
2. go to Hue in task bar.

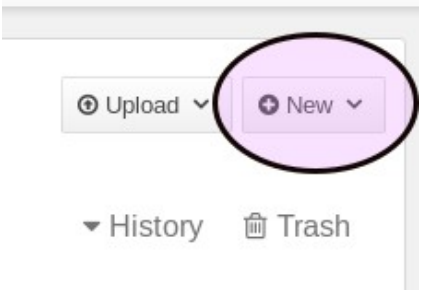


3. Go to File Browser

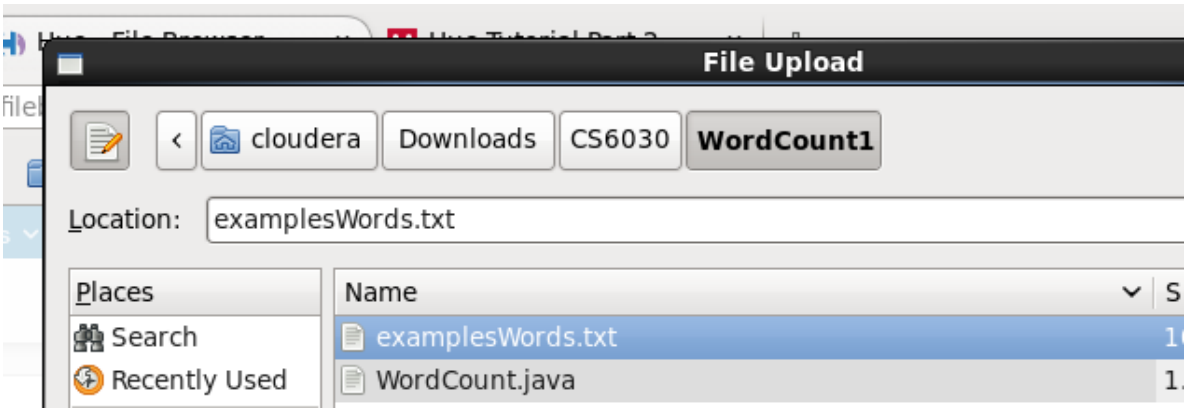


4. from the folders that displayed click on: 

5.You will reach Oozie folder, click on this folder, and then click on ***New*** and then on ***Directory***. Give a name for your directory, let say WordCount



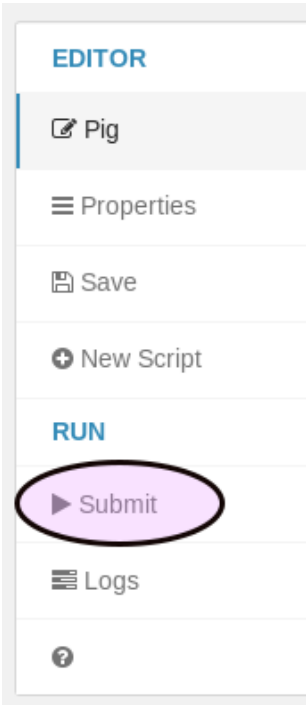
6. click on your new folder and click on Upload next to New in previous figure. Now upload the file that you want to count the word on it. Let say we will work on the same file that we have from our CS6030 class.



7. Now, go to Query Editor, in the above task bar and select Pig from the drop down meny.

8. in the editor that you will see copy and paste the following Pig Latin code:

```
A = LOAD '/oozie/wordcount' USING TextLoader() AS (words:chararray);
B = FOREACH A GENERATE FLATTEN(TOKENIZE(*));
C = GROUP B BY $0;
D = FOREACH C GENERATE group, COUNT(B);
STORE D INTO '/oozie/wcresults';
```



9. click on Submit in the Editor in left side.
10. Click on Save to save the script.
11. to see the result click on File Browser and then go to Oozie folder you will see their a folder called named wcresult. Click on this folder to see the result, it will be some thing like figure below.