COLLECTING TWEETS USING TWITTER STREAMING API'S

SUBJECT: PRINCIPLES OF THE BIG DATA MANAGEMENT

PROJECT PHASE1

SESHA SAI SRIKAR PAVAN KUMAR CHONGALA (16273361)

Gowtham Varma Vegnesna (16286291)

Goutham Gandreddi (16286424)

INSTRUCTOR

Dr. PRAVEEN RAO, Ph.D.

ABSTRACT

This project collects tweets from Twitter using Twitter API. The collected tweets in JSON format are pushed into HDFS and the hashtags and URLs are extracted using python code. The word count program is then run on the extracted hashtags and URL's document both in Hadoop and spark.

DISCRIPTION OF THE PROBLEM:

Nowadays, data is growing and accumulating faster than ever before. Currently, around 90% of all data generated in our world was generated only in the last two years. Due to this staggering growth rate, big data platforms had to adopt radical solutions to maintain such huge volumes of data.

Data collection from social networking sites can help analyzing easily. As data collected is unstructured. Hadoop, Spark and python are the best big data solutions.

HADOOP DISTRIBUTED FILE SYSTEM (HDFS):

Big data uses Hadoop Distributed File System (HDFS) which can store Petabytes of data. It can store both structured and unstructured data. It uses social media for survey. The proposed system is totally an automatic system. It uses the social media data to conduct a survey and generates the same result as it has done manually. It has the following advantages:

- Data is available in real time and at high frequency.
- Low-cost source of valuable information.
- No wastage of manpower and time.
- It can answer the questions that would have known through surveys in advance.
- Social media offer a distinctive window into economic activity

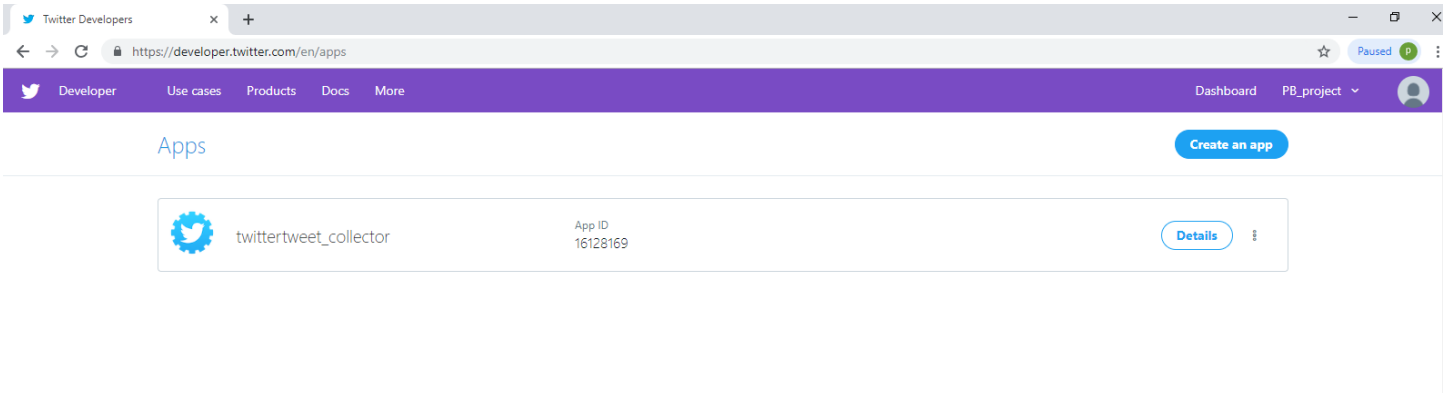
System Configuration:

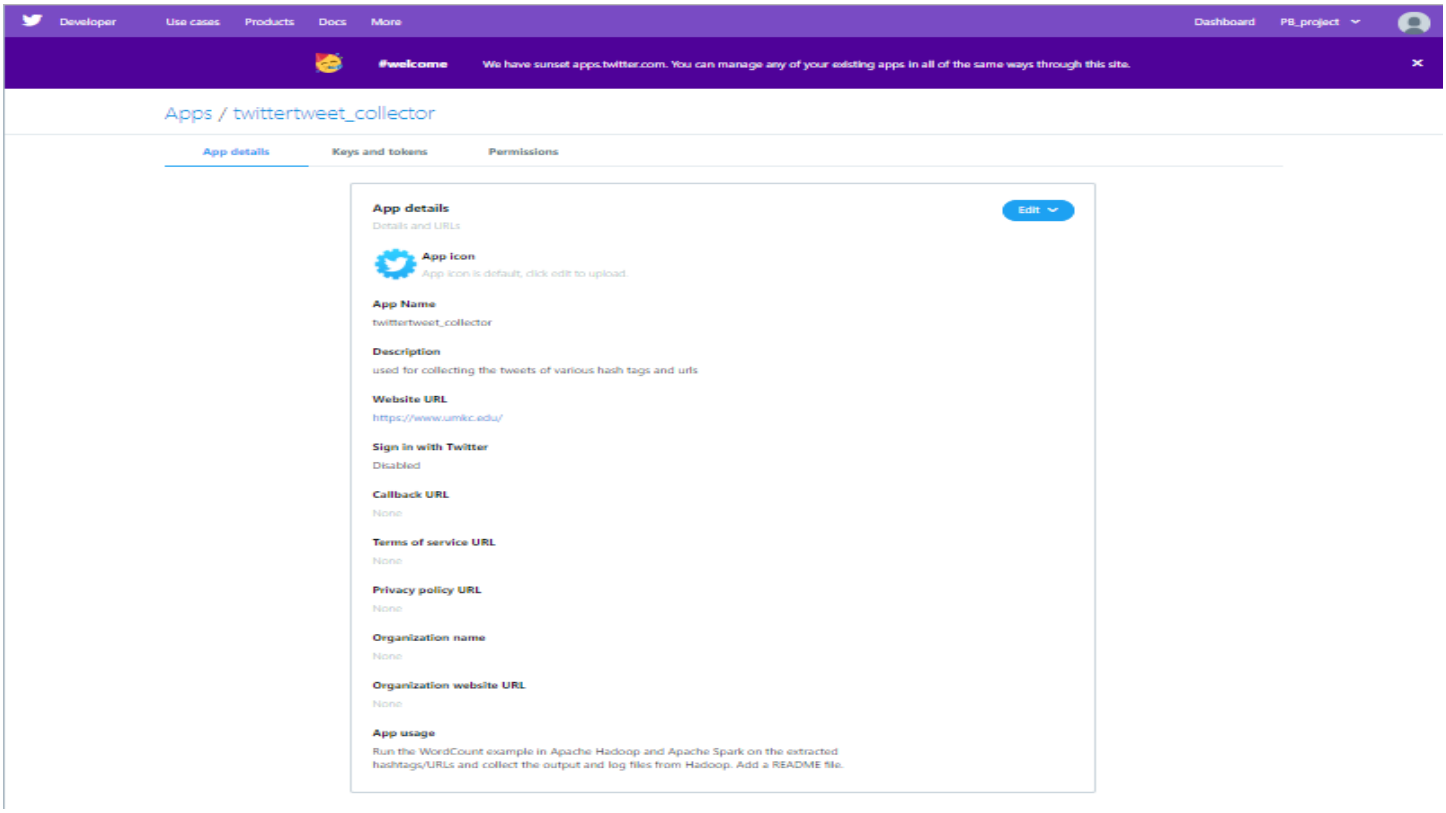
- Operating System:
  Mac OS - 10.14.3
- Tools used:
  Hadoop 3.1.2
  Spark 2.4
- Language:
  Python 3

Creating Credentials for Twitter APIs

Before collection of tweets, it is important to sign in with twitter and signup for a development account with the following link: https://dev.twitter.com/resources/signup

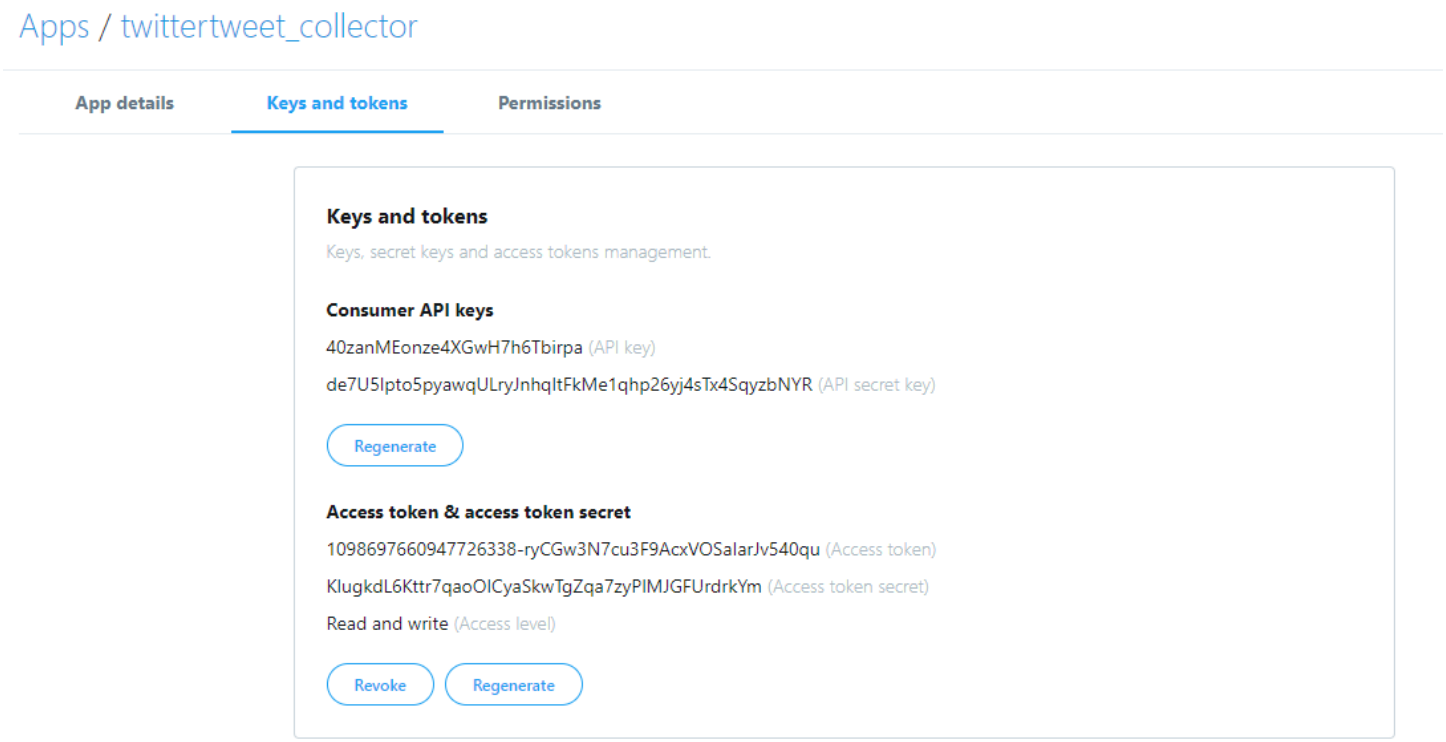
Create a Twitter Application “twittertweet_collector” with these we get 4 keys – Consumer key, Consumer secret key, Access token and Access token secret which are later used for the collection of tweets

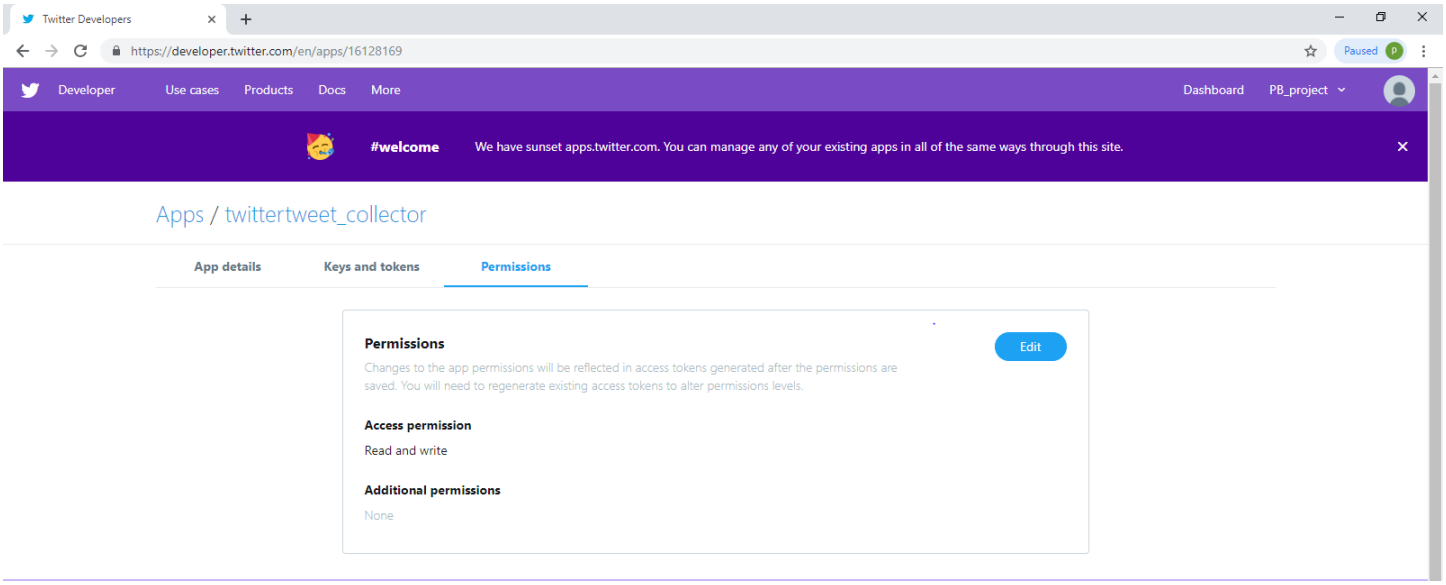




Keys and tokens:

Consumer key, Consumer secret, Access token and Access token secret keys which are used for fetching the tweets from the twitter

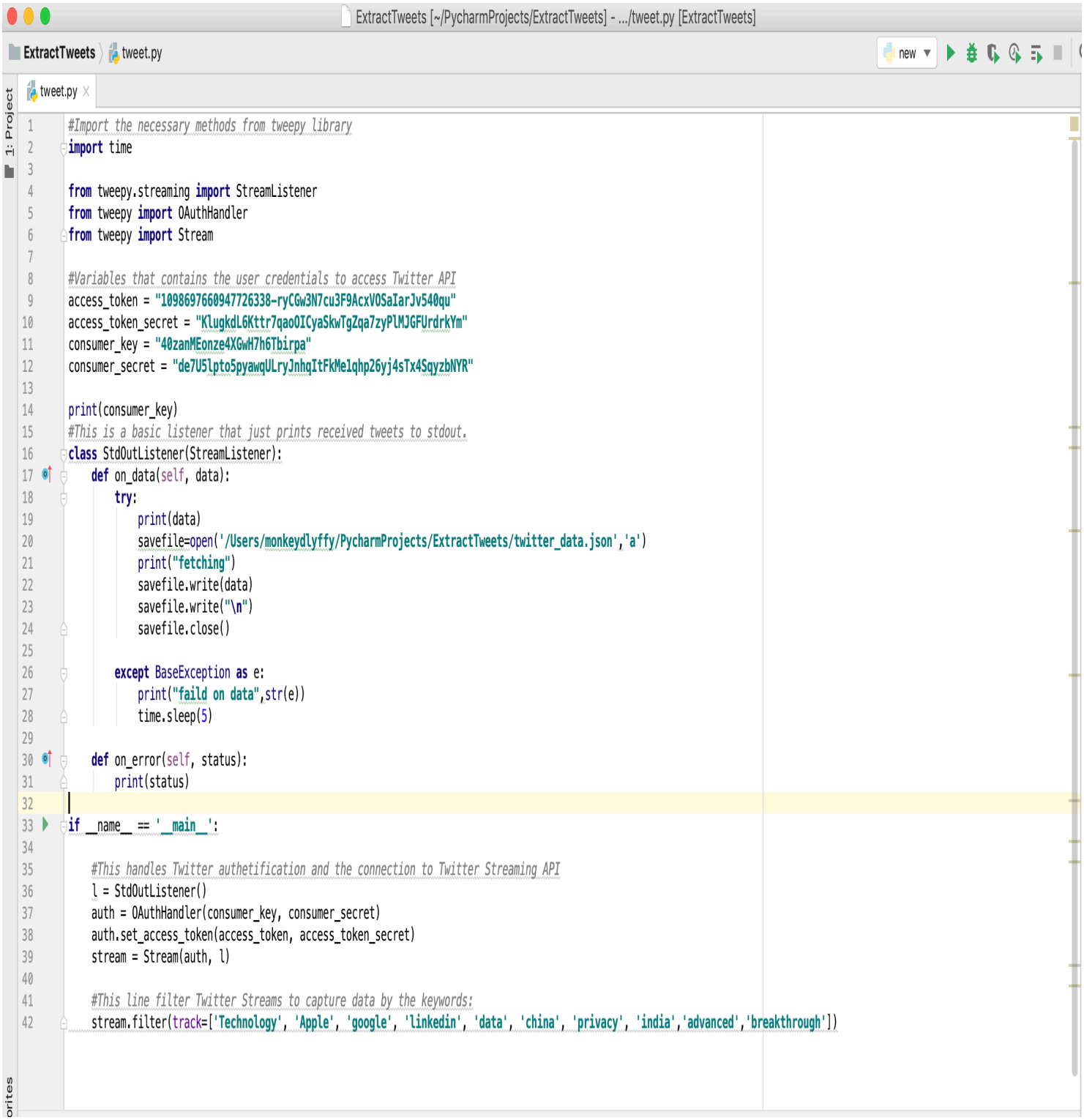




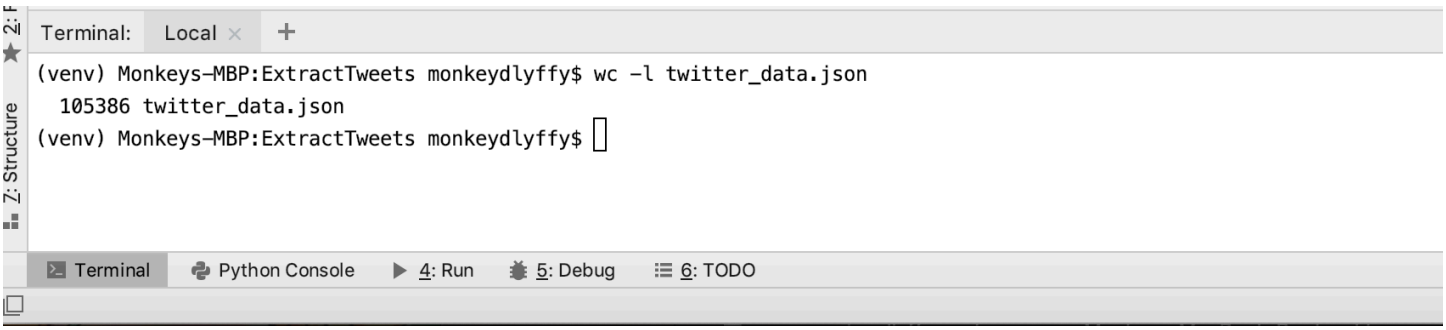
Twitter API:

Twitter streaming API is used for downloading large volume of tweets from the twitter using python 3

Code for collecting tweets

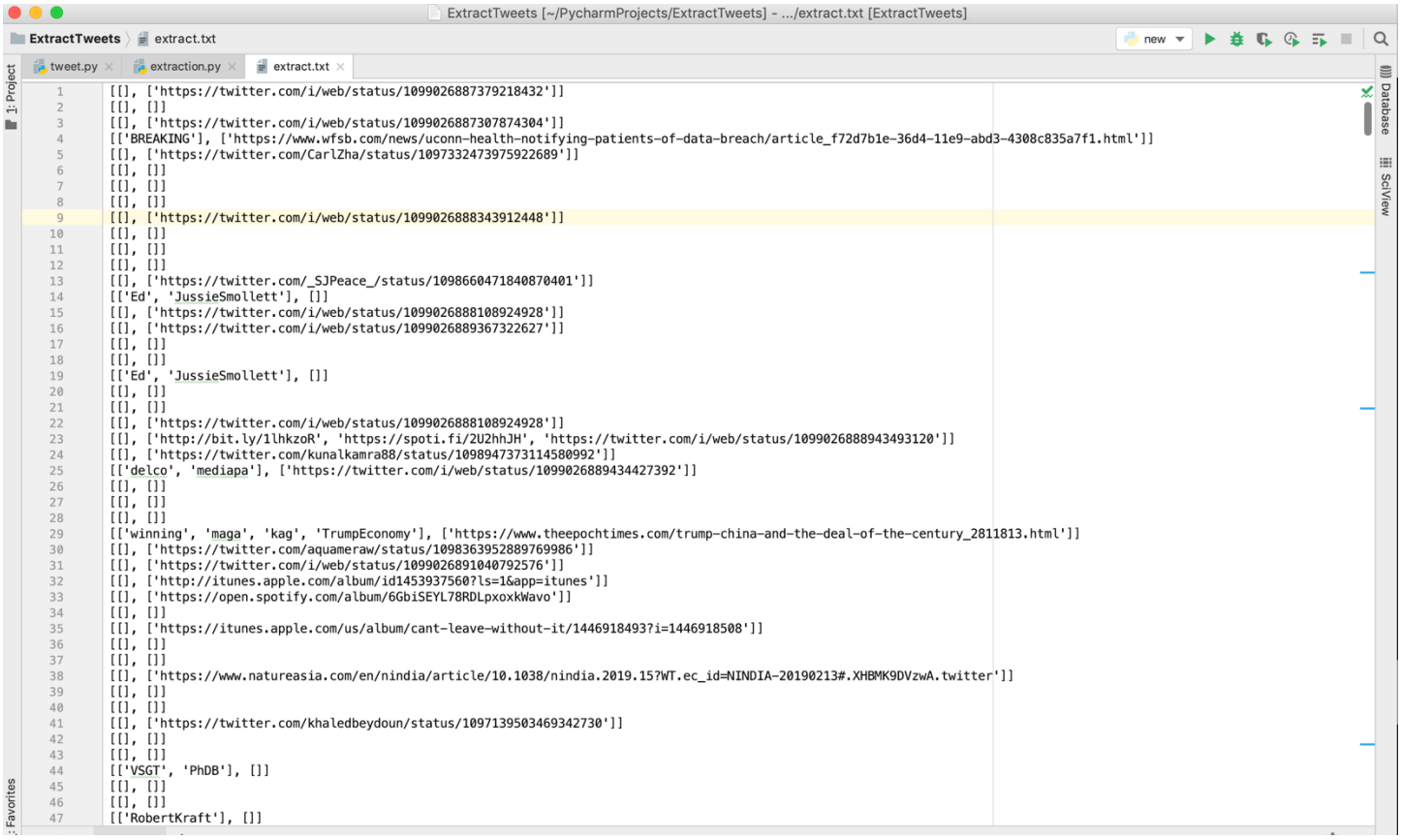


Number of tweets downloaded to twitter_data.json file



Fetching Code for extracting Hashtags and Url's from Tweets

After extracting the hashtags and url’s

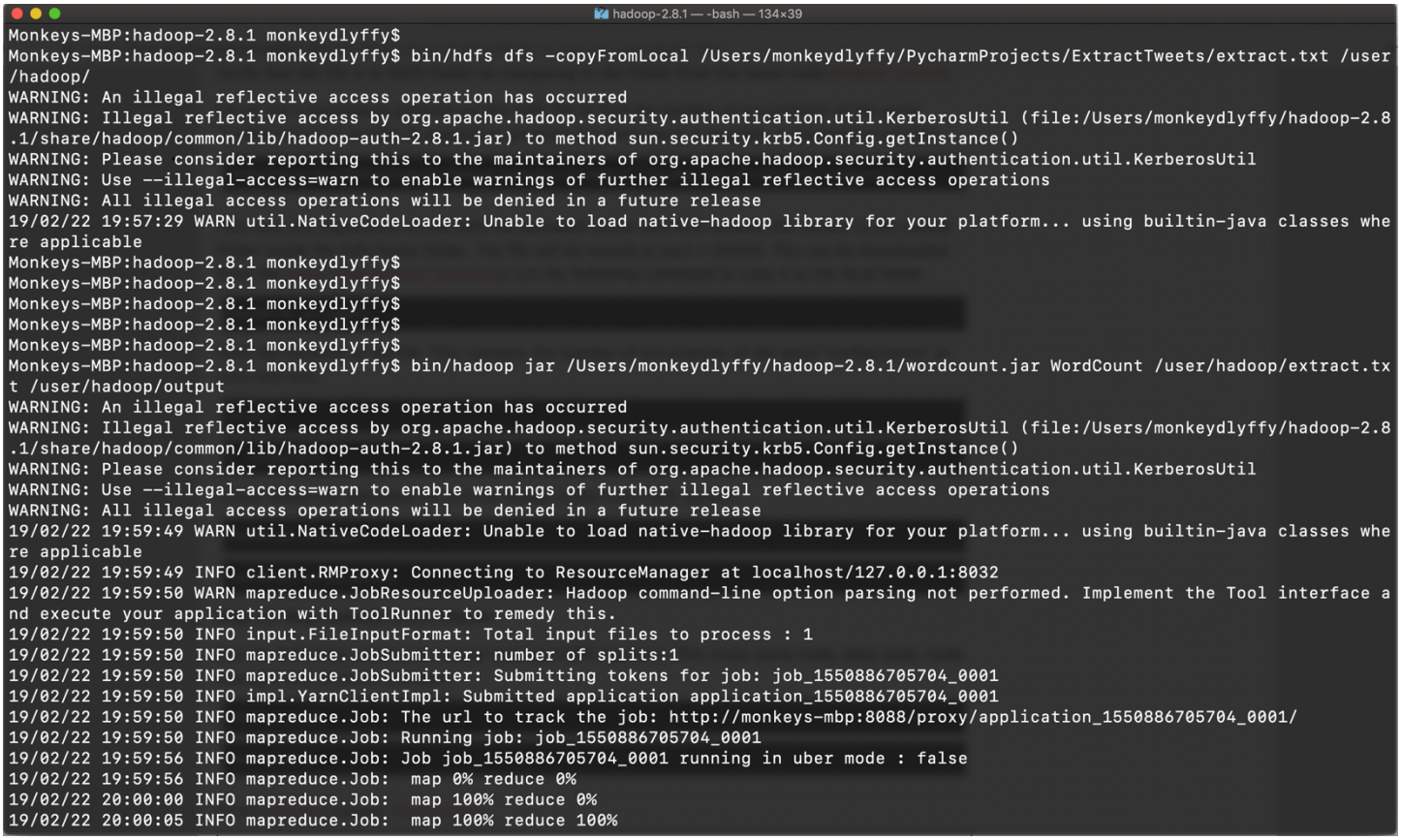


Runing The Word Count for the gathered tweets –

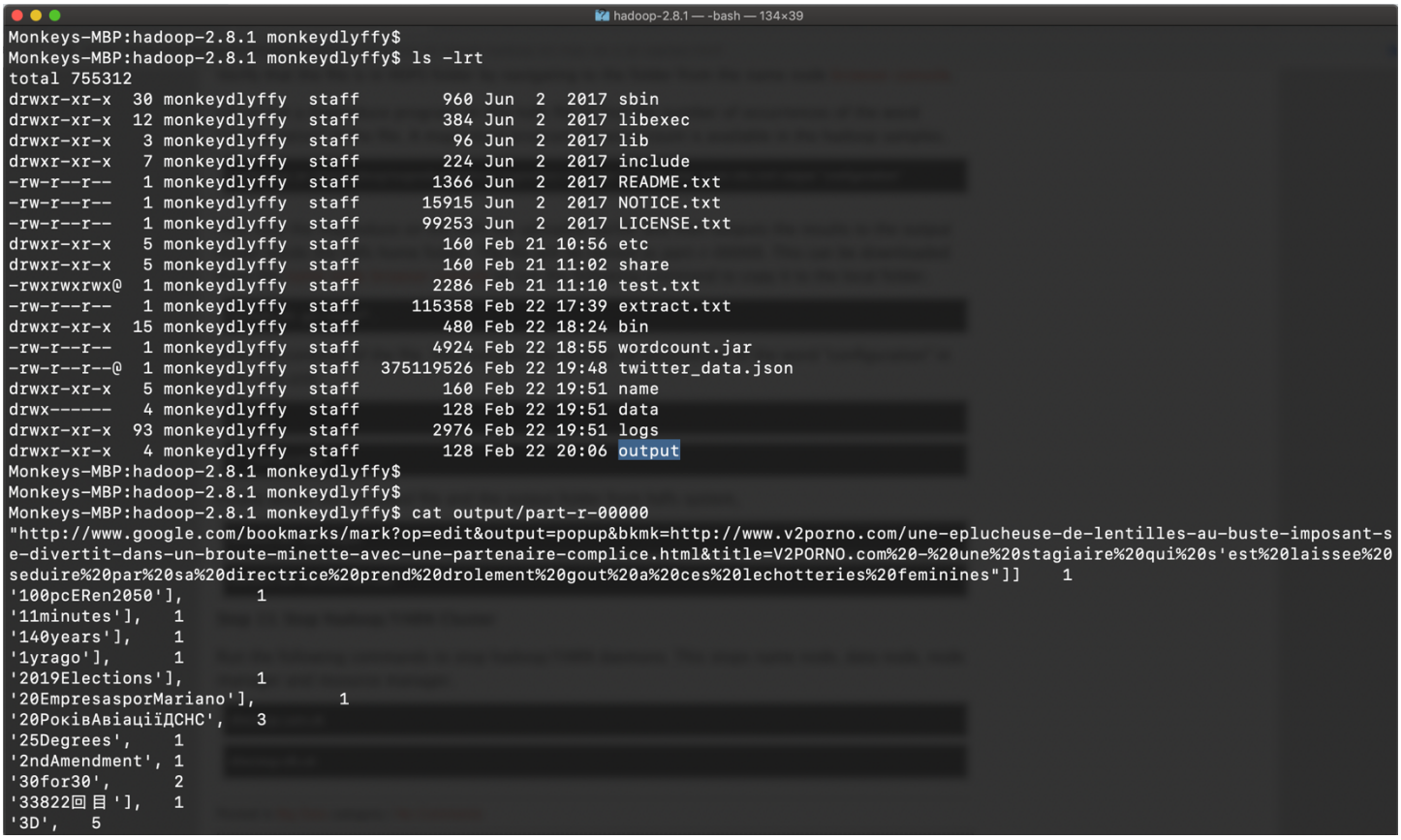
Code for word count –

```
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);
  }
}
```

Pushing the extracted file & running the wordcount program



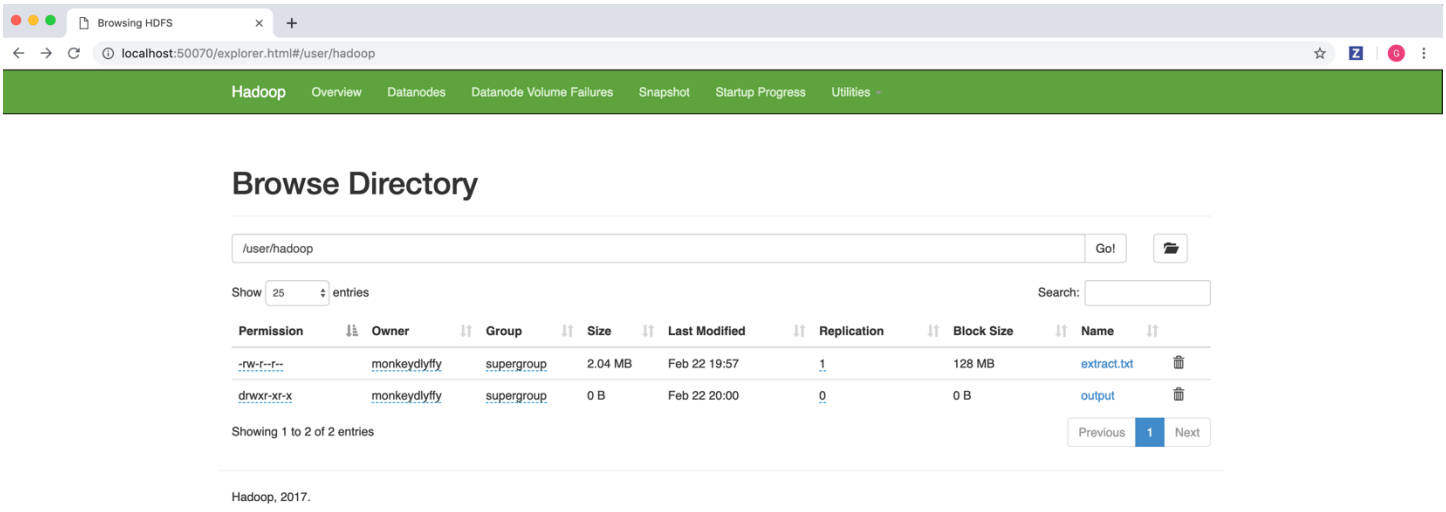
Copied the output folder to local and viewing the contents of part-r-0000 file:



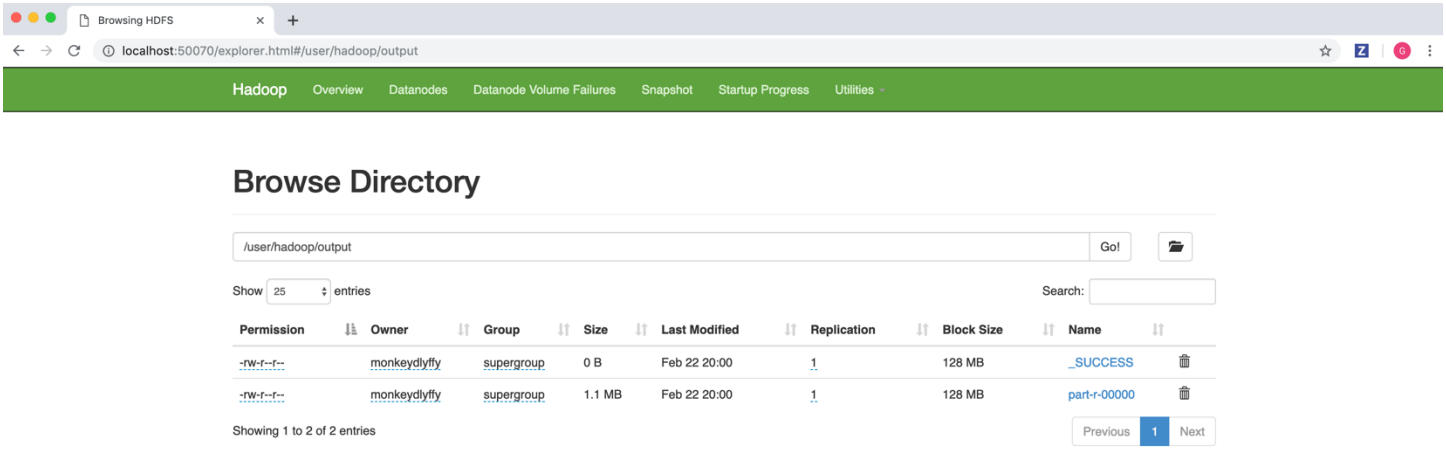
word count output:

Screenshots from HDFS:

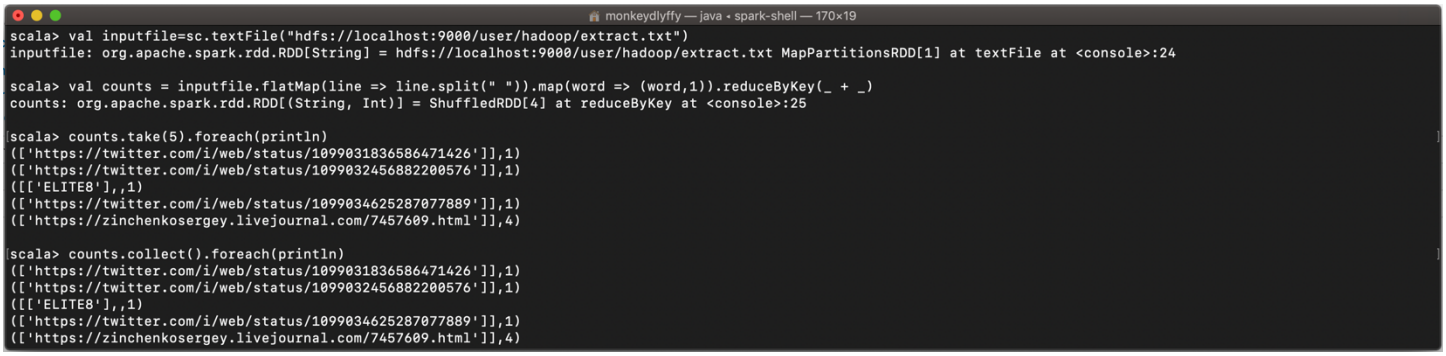
Before running the wordcount:



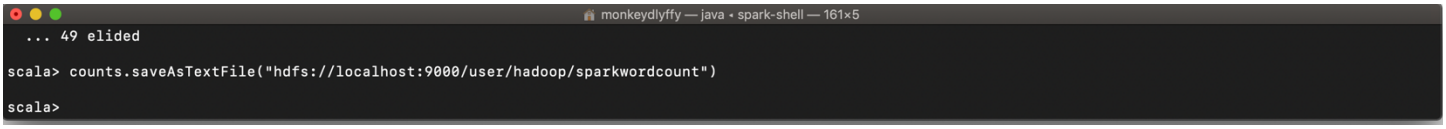
After the wordcount - new directory output creation & its contents:



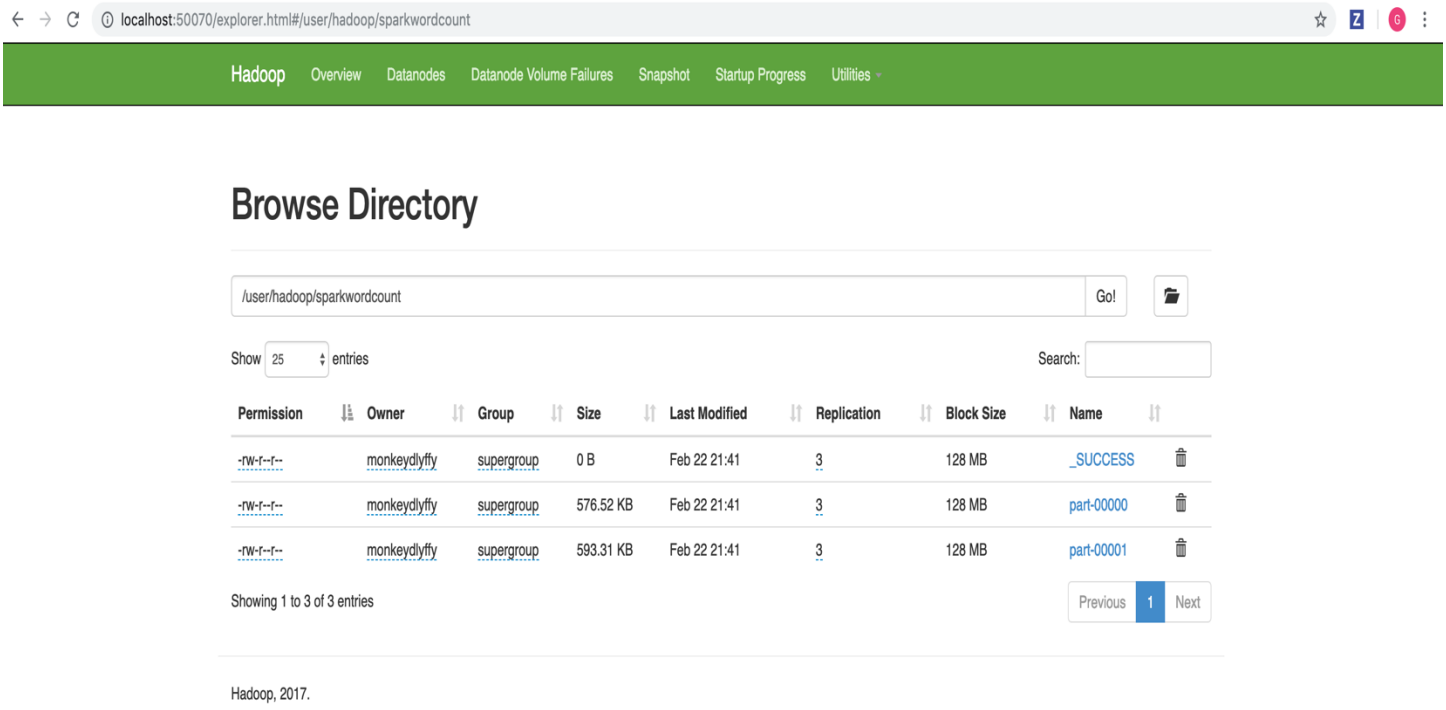
Code for running the word count in spark

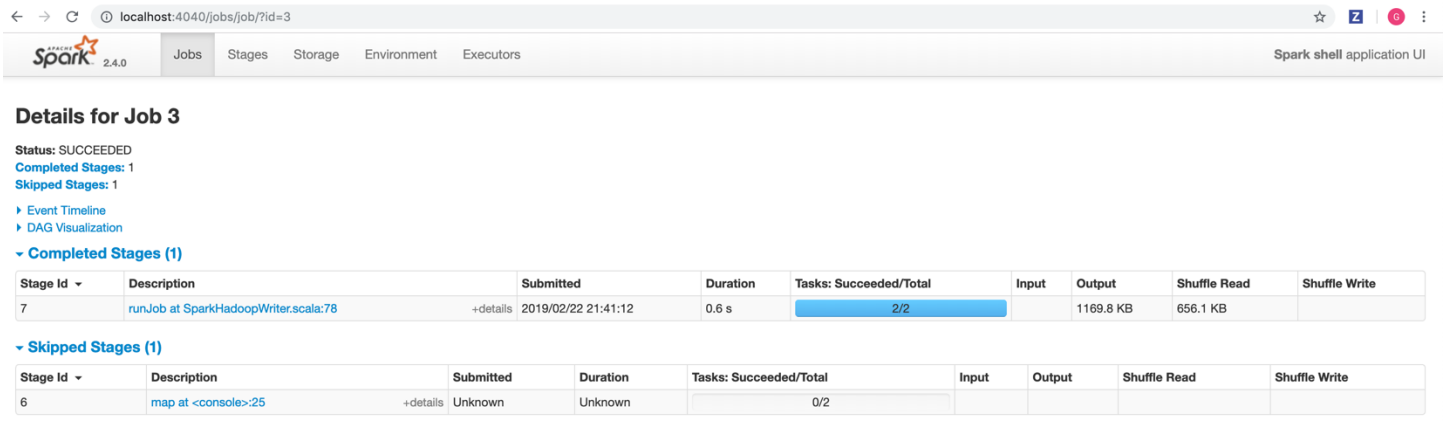


New output directory for spark wordcount:



Screenshots from HDFS for spark:





The log files and the readme file can be found in the following location:

https://drive.google.com/drive/u/0/folders/149xQxhQwCcDJThNJnwyjc85qtM8AKj02

References:

1) https://www.alexkras.com/how-to-get-user-feed-with-twitter-api-and-python/
2) http://socialmedia-class.org/twittertutorial.html
3) https://spark.apache.org/examples.html