**BIG DATA ANALYTICS REPORT**

**Analyzing Data from Twitter**

**Hadoop**

As organizations are getting flooded with massive amount of raw data, the challenge here is that traditional tools are poorly equipped to deal with the scale and complexity of such kind of data. That's where Hadoop comes in. Hadoop is well suited to meet many Big Data challenges, especially with high volumes of data and data with a variety of structures. Hadoop is a framework for storing data on large clusters of commodity hardware, everyday computer hardware that is affordable and easily available and running applications against that data. A cluster is a group of interconnected computers (known as nodes) that can work together on the same problem. Hadoop consists of two main components:

1) HDFS (Data Storage)

2) Map-Reduce (Analyzing and Processing)

**Task 1: Getting tweets and saving them into HDFS**

**Creating Twitter Application**

First of all if we want to do sentiment analysis on Twitter data we want to get Twitter data first so to get it we want to create an account in Twitter developer and create an application.   
After creating, we have the API key, the API secret, the access token, and the access token secret. Create flume.conf directory by copying from flume.conf.template.We open flume.conf file in the directory /usr/lib/flume/conf and then change the following keys in the file. These keys will be obtained from the page above. These are the keys which we will change in the flume.conf file: Access Token, Access Token Secret, Consumer Key (API Key), Consumer Secret (API Secret) Also add the keywords that we want to extract from twitter. Here, we are extracting data on Ankara. After creating an application in the Twitter developer site, we can now access the Twitter and we can get the information that we want. Here we will get everything in JSON format and this is stored in the HDFS that we have given the location where to save all the data that comes from the Twitter. After running the Flume, the Twitter data will automatically will save into HDFS. After creating an application in the Twitter developer site, we can now access the Twitter and we can get the information that we want. Here we will get everything in JSON format and this is stored in the HDFS that we have given the location where to save all the data that comes from the Twitter. After running the Flume, the Twitter data will automatically will save into HDFS.

**Getting Data using Flume**

After creating an application in the Twitter developer site, we can now access the Twitter and we can get the information that we want. Here we will get everything in JSON format and this is stored in the HDFS that we have given the location where to save all the data that comes from the Twitter. After running the Flume, the Twitter data will automatically will save into HDFS.

-Open the terminal and start all the services and add flume service.

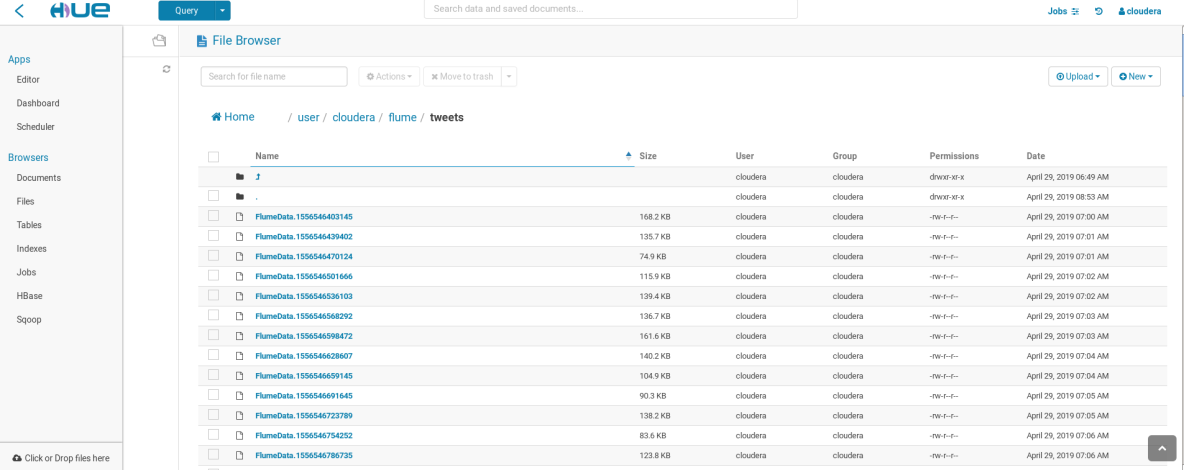
-Download the flume-sources-1.0-SNAPSHOT.jar. The jar contains the java classes to pull the Tweets and save them into HDFS.(put flume-sources-1.0-SNAPSHOT.jar file into /usr/lib/flume-ng/lib directory)

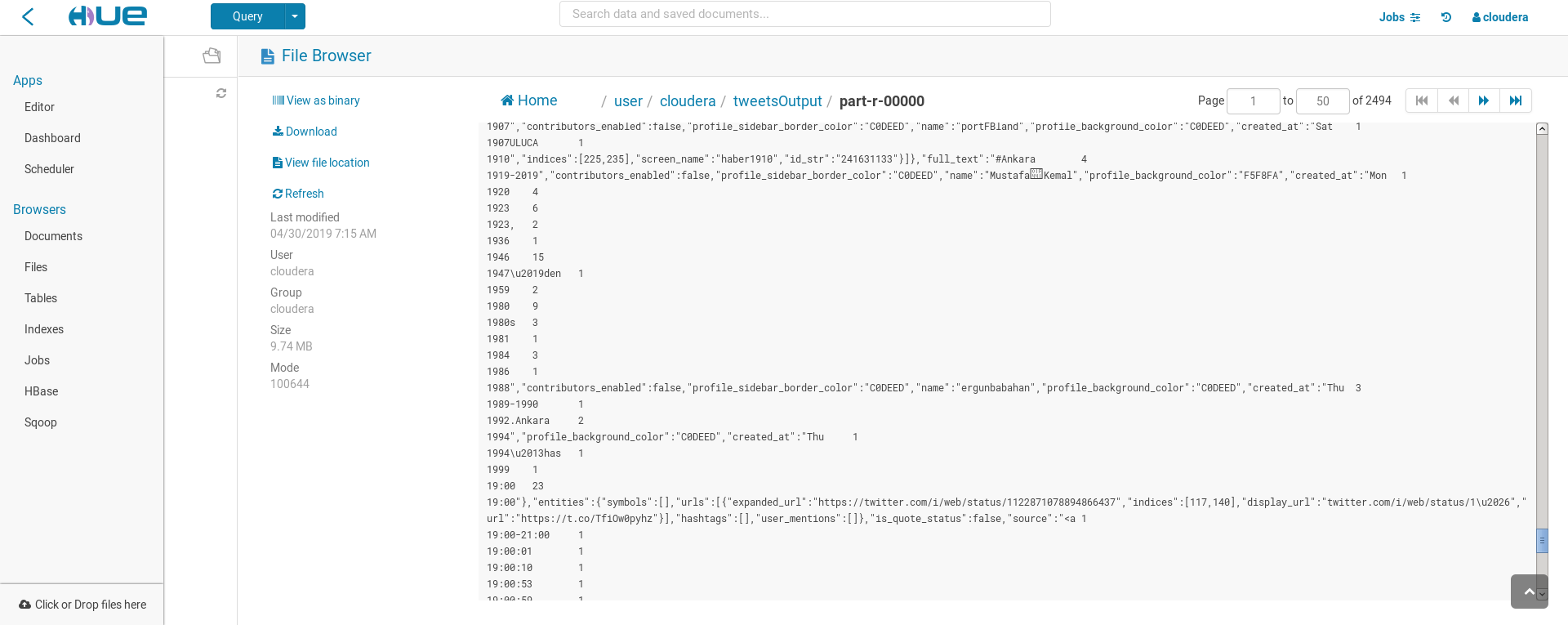
-We must also add flume\_path into flume-env.sh file and this path is flume-sources-1.0-SNAPSHOT.jar’s path.

-Update all twitter4j to last version and change their names.

-We will now start the flume agent using the following command: flume-ng agent --conf ./conf/ -f conf/flume.conf -Dflume.root.logger=DEBUG,console -n TwitterAgent in /usr/lib/flume-ng/conf directory (flume.conf is in different conf directory which is in /usr/lib/flume-ng/conf directory)

-This is the list of twitter data extracted which contains the keyword as specified in the conf file.

****

We have collected and stored the tweets in HDFS using Flume in the previous section. The tweets are located in the following location of the HDFS: TwitterAgent.sinks.HDFS.hdfs.path = hdfs://localhost:8020/user/cloudera/flume/tweets

**Task 2: Analysis of the tweet**

Java file opened from eclipse. Added external libraries for mapreduce. The mapreduce code home / cloudera path was executed as executable jar file. Map Reduce code:

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]));

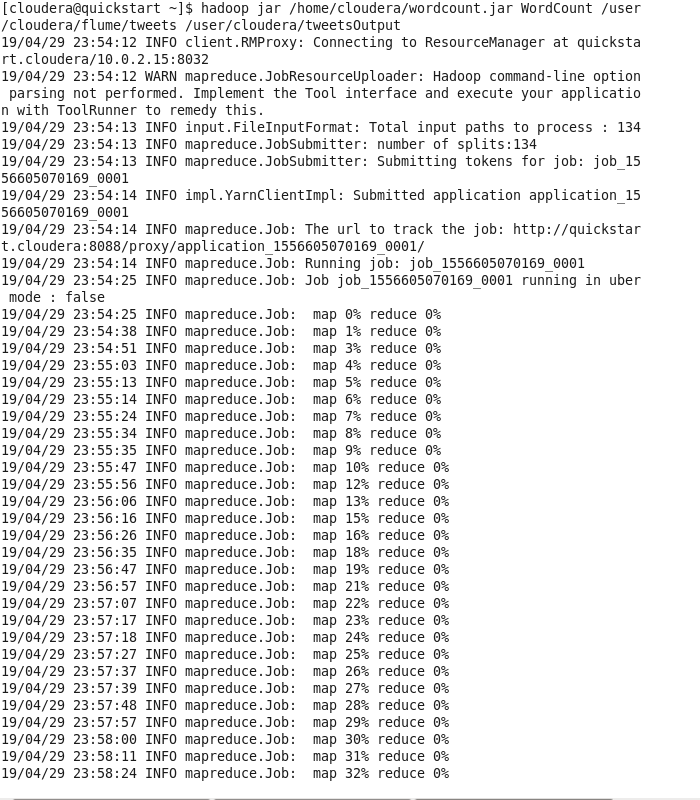
System.exit(job.waitForCompletion(true) ? 0 : 1);

}

}

In the code as the input to the output of our tweet data as the output file we want to create the way was given.

After that, the code specified in the picture was executed



Afterwards, the hour data was analyzed from the obtained data and our table posting tweets about Ankara in 12 Hour from 8.00 am to 8.00 pm was created in the excel file.