Assignment-5 README

1. Spark Streaming with Twitter and Kafka:

Created a spark stream application- TwitterStreaming

We created a twitter developer account and fed the credentials to .properties file which was then imported in the main class to access, the streaming data of the topics provided.

The following steps were performed to run the file locally.

Starting Zookeeper  
- cd dev/zookeeper-3.4.14/  
- bin/zkServer.sh start  
  
- bin/zkCli.sh -server [127.0.0.1:2181](http://127.0.0.1:2181)  
\* Zookeeper will now be running on one of the terminals  
  
2. Starting Kafka  
- cd dev/kafka\_2.12-2.2.0/  
- bin/kafka-server-start.sh config/server.properties  
\* Now Kafka will be running on the other terminal  
  
3. Run Program after packaging into a jar  
- /home/mr/dev/spark-2.4.1-bin-hadoop2.7/bin/spark-submit --class streaming.TwitterProducer.scala twitter-1.0-jar-with-dependencies.jar Basketball "NBA"  
  
4. Visualization  
- cd elasticsearch-7.0.0/  
- ./bin/elasticsearch  
- curl localhost:9200  
- ./bin/kibana  
- .bin/logstash -f logstash-simple.conf

Output: It outputs the sentiment of the tweet as:

1. Positive
2. Negative
3. Neutral

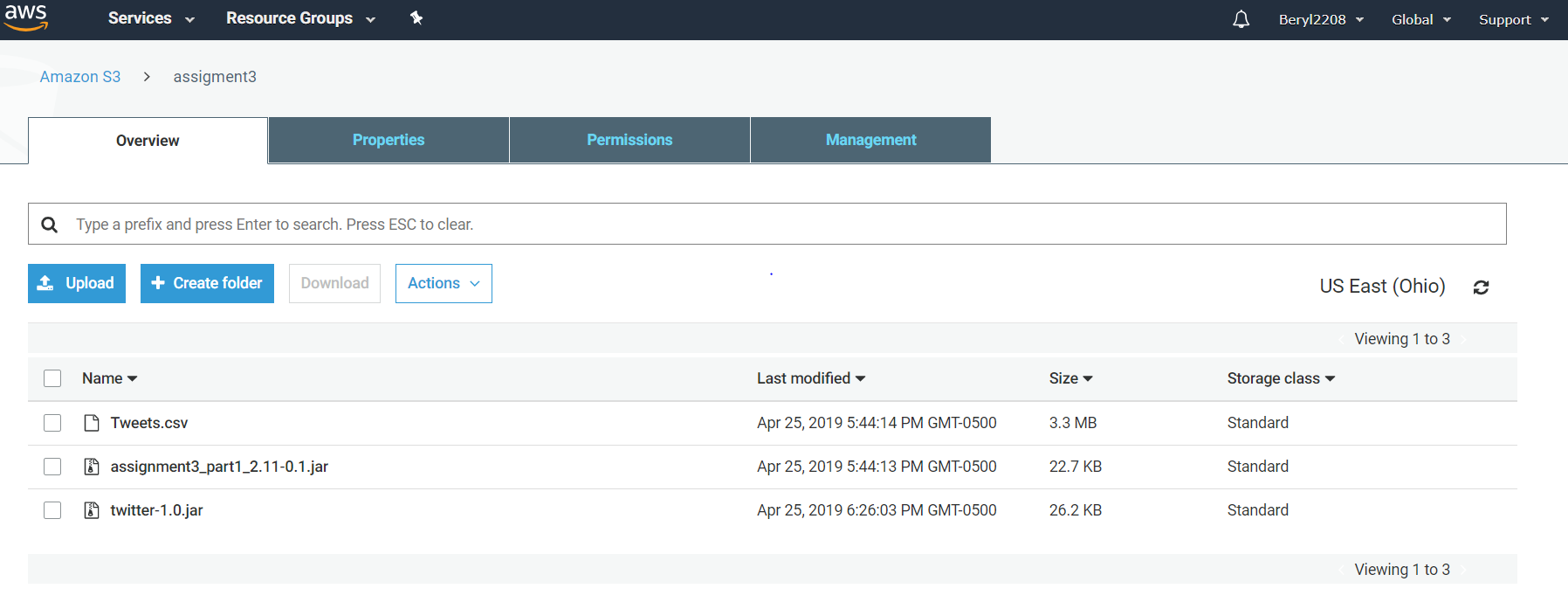
We’ve made use of the core StanfordNLP library to create a pipeline and preprocess the text and sentences.

The exception for empty tweets is also handled. However, we were unsuccessful in visualizing it in Kibana.

1. Tweet Processing & Classification using Pipelines:

Created a Scala class, as a part of a Scala SBT project with all the dependencies needed to be run on AWS. Compiled and packaged the code into a jar.

Input file and jar uploaded to AWS S3 and run on EMR instance.



After creating a cluster, we add steps and run jar as a spark application by passing the arguments namely the tweets.csv and the jar-path as in the s3 bucket.

Successfully created output files in output folder in AWS S3 bucket after running the jar.

