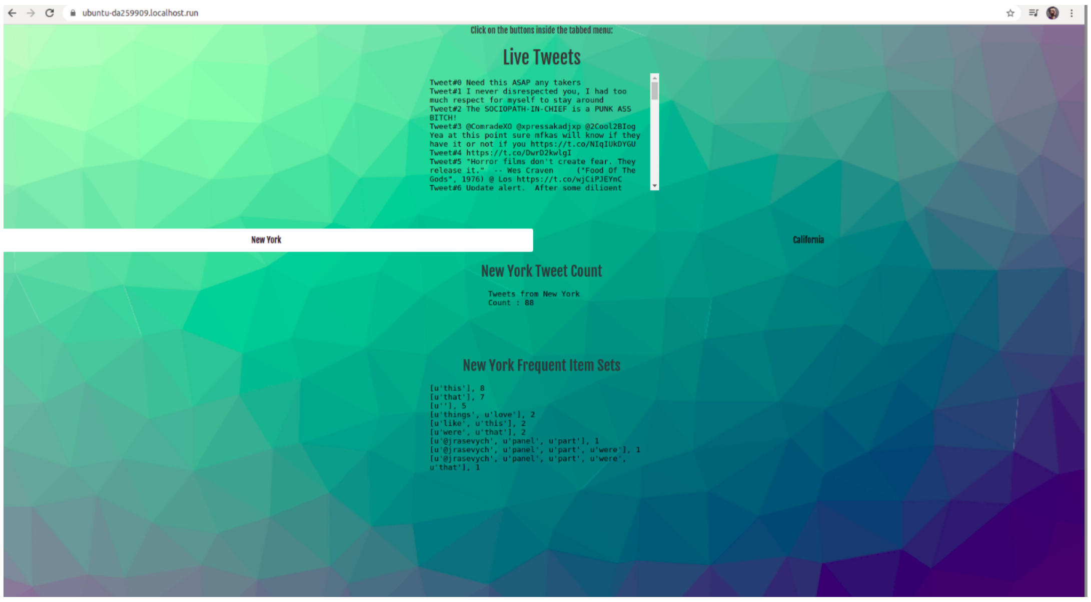
CS6847 Cloud Computing. Assignment - 5

Suhas Pai, CS17B116

1 Setup

I installed Apache Kafka, Apache Spark, and Zookeeper, initially on a single machine, and created its AMI Image. Zookeeper is initially started, and then the Kafka server is started. The config files are modified on both instances, to let the zookeeper server know the IP addresses of the Kafka brokers. The producer config file is changed as well. Zookeeper server is run only on one instance, while the Kafka server is started on both instances. Then, on one of the instances, a python script is run. The script uses Tweepy and PyKafka. Tweepy is used for collecting all tweets from New York and Los Angeles ,based on the location coordinates of the tweet. Two producers are created using PyKafka, and these tweets are then sent to the broker. Each instance holds one topic's data. Then, the Spark master is started. The Streaming Context is then started, and a stream is created. Tweets are then picked up from the appropriate topic, written to a file, and then the mllib library is used to run the frequent dataset algorithm. A http server is run in the folder of the html index file, and then using a python tool, the local website is made global.


ubuntu-da259909.localhost.run
Click on the buttons inside the tabbed menu:
Live Tweets
New York
California
New York Tweet Count
New York Frequent Item Sets

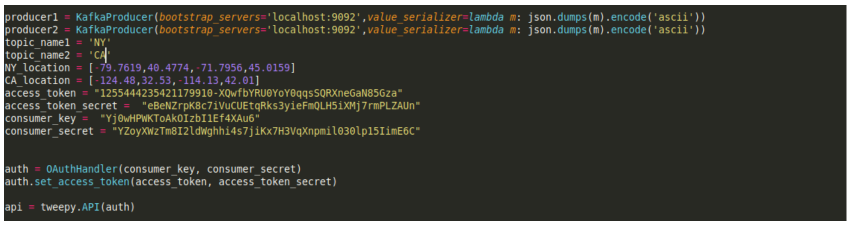
```
Tweet#0 Need this ASAP any takers
Tweet#1 I never disrespected you, I had too
much respect for myself to stay around
Tweet#2 The SOCIOPATH-IN-CHIEF is a PUNK ASS
BITCH!
Tweet#3 @ComradeXO @xpressakadjxp @2Cool2BIog
Yea at this point sure mfkas will know if they
have it or not if you https://t.co/NIqIUkDYGU
Tweet#4 https://t.co/DwrD2kwlgI
Tweet#5 "Horror films don't create fear. They
release it."  -- Wes Craven     ("Food Of The
Gods", 1976) @ Los https://t.co/wjCiPJEYnC
Tweet#6 Update alert.  After some diligent
```

```
Tweets from New York
Count : 88
```

```
[u'this'], 8
[u'that'], 7
[u''], 5
[u'things', u'love'], 2
[u'like', u'this'], 2
[u'were', u'that'], 2
[u'@jrasevych', u'panel', u'part'], 1
[u'@jrasevych', u'panel', u'part', u'were'], 1
[u'@jrasevych', u'panel', u'part', u'were',
u'that'], 1
```

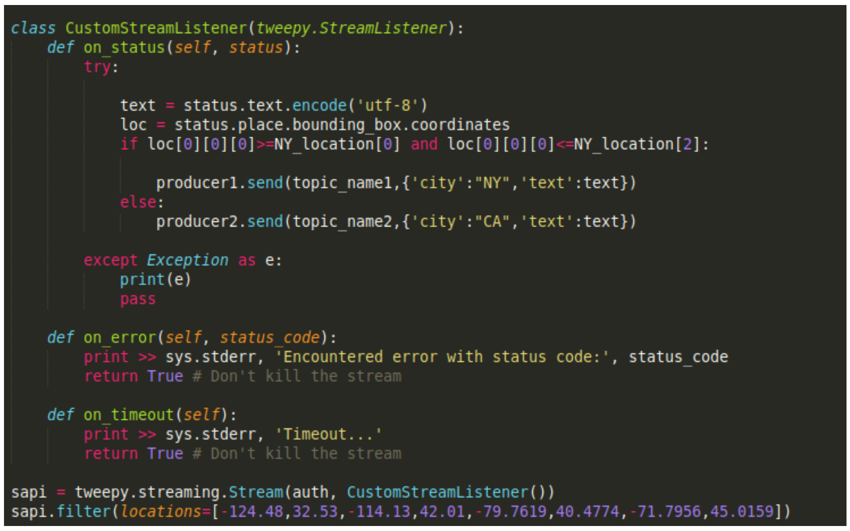
Final website

2 Generating Tweets



```
producer1 = KafkaProducer(bootstrap_servers='localhost:9092',value_serializer=lambda m: json.dumps(m).encode('ascii'))
producer2 = KafkaProducer(bootstrap_servers='localhost:9092',value_serializer=lambda m: json.dumps(m).encode('ascii'))
topic_name1 = 'NY'
topic_name2 = 'CA'
NY_location = [-79.7619,40.4774,-71.7956,45.0159]
CA_location = [-124.48,32.53,-114.13,42.01]
access_token = "1255444235421179910-XQwfbYRU0YoY0qqsSQRXneGaN85Gza"
access_token_secret =  "eBeNZrpK8c7iVuCUEtqRks3yieFmQLH5iXMj7rmPLZAUn"
consumer_key =  "Yj0wHPWKToAkOIzbI1Ef4XAu6"
consumer_secret = "YZoyXWzTm8I2ldWghhi4s7jiKx7H3VqXnpmil030lp15IimE6C"


auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)

api = tweepy.API(auth)
```

Initial setup for tweepy and PyKafka



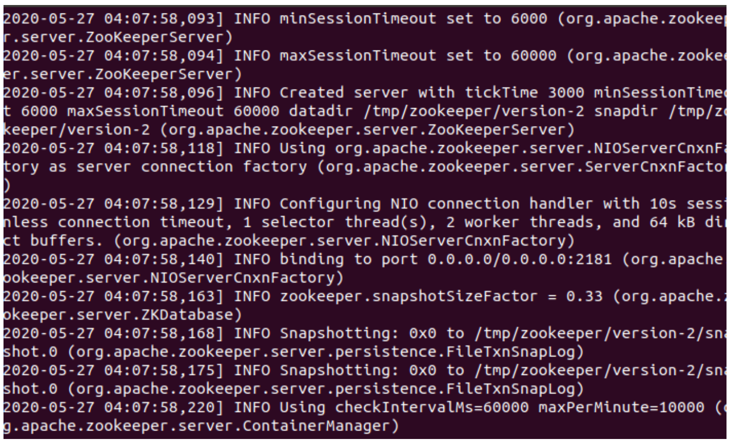
```
class CustomStreamListener(tweepy.StreamListener):
    def on_status(self, status):
        try:

            text = status.text.encode('utf-8')
            loc = status.place.bounding_box.coordinates
            if loc[0][0][0]>=NY_location[0] and loc[0][0][0]<=NY_location[2]:

                producer1.send(topic_name1,{'city':"NY",'text':text})
            else:
                producer2.send(topic_name2,{'city':"CA",'text':text})

        except Exception as e:
            print(e)
            pass

    def on_error(self, status_code):
        print >> sys.stderr, 'Encountered error with status code:', status_code
        return True # Don't kill the stream

    def on_timeout(self):
        print >> sys.stderr, 'Timeout...'
        return True # Don't kill the stream

sapi = tweepy.streaming.Stream(auth, CustomStreamListener())
sapi.filter(locations=[-124.48,32.53,-114.13,42.01,-79.7619,40.4774,-71.7956,45.0159])
```

The class that filters tweets based on Box coordinates of New York and California

Two producers are started using PyKafka, one for each topic. The zookeeper ip address is the default address of localhost, and the tweets are dumped in json format, so that they can be filtered easily in the stream. Every tweet is read, and based on which box the coordinate falls in, it's location is identified.

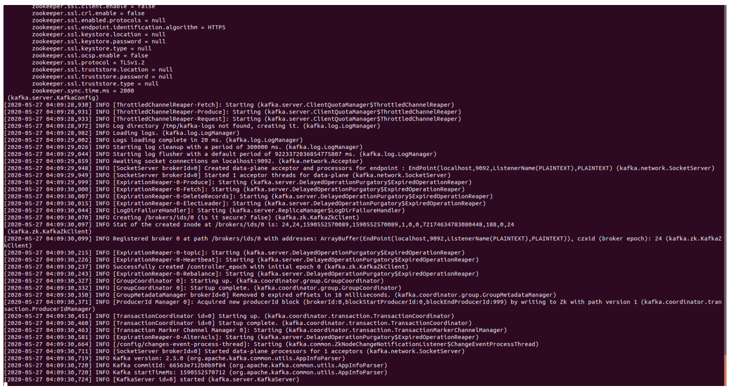
3 Starting Kafka and Zookeeper

Zookeeper is started on one of the instances. Then two brokers, with different broker ids, are started on the two instances.



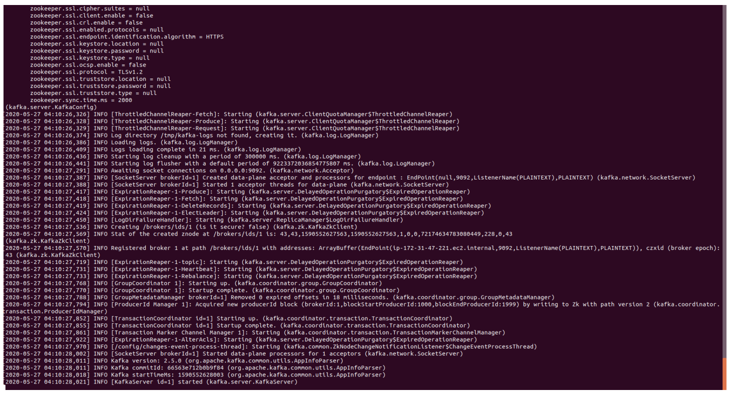
```
2020-05-27 04:07:58,093] INFO minSessionTimeout set to 6000 (org.apache.zookee
r.server.ZooKeeperServer)
2020-05-27 04:07:58,094] INFO maxSessionTimeout set to 60000 (org.apache.zooke
er.server.ZooKeeperServer)
2020-05-27 04:07:58,096] INFO Created server with tickTime 3000 minSessionTime
t 6000 maxSessionTimeout 60000 datadir /tmp/zookeeper/version-2 snapdir /tmp/z
keeper/version-2 (org.apache.zookeeper.server.ZooKeeperServer)
2020-05-27 04:07:58,118] INFO Using org.apache.zookeeper.server.NIOServerCnxnF
tory as server connection factory (org.apache.zookeeper.server.ServerCnxnFacto
)
2020-05-27 04:07:58,129] INFO Configuring NIO connection handler with 10s sess
nless connection timeout, 1 selector thread(s), 2 worker threads, and 64 kB di
ct buffers. (org.apache.zookeeper.server.NIOServerCnxnFactory)
2020-05-27 04:07:58,140] INFO binding to port 0.0.0.0/0.0.0.0:2181 (org.apache
ookeeper.server.NIOServerCnxnFactory)
2020-05-27 04:07:58,163] INFO zookeeper.snapshotSizeFactor = 0.33 (org.apache.
okeeper.server.ZKDatabase)
2020-05-27 04:07:58,168] INFO Snapshotting: 0x0 to /tmp/zookeeper/version-2/sn
shot.0 (org.apache.zookeeper.server.persistence.FileTxnSnapLog)
2020-05-27 04:07:58,175] INFO Snapshotting: 0x0 to /tmp/zookeeper/version-2/sn
shot.0 (org.apache.zookeeper.server.persistence.FileTxnSnapLog)
2020-05-27 04:07:58,220] INFO Using checkIntervalMs=60000 maxPerMinute=10000 (
g.apache.zookeeper.server.ContainerManager)
```

Zookeeper being started



```
        zookeeper.ssl.client.enable = false
        zookeeper.ssl.crl.enable = false
        zookeeper.ssl.enabled.protocols = null
        zookeeper.ssl.endpoint.identification.algorithm = HTTPS
        zookeeper.ssl.keystore.location = null
        zookeeper.ssl.keystore.password = null
        zookeeper.ssl.keystore.type = null
        zookeeper.ssl.ocsp.enable = false
        zookeeper.ssl.protocol = TLSv1.2
        zookeeper.ssl.truststore.location = null
        zookeeper.ssl.truststore.password = null
        zookeeper.ssl.truststore.type = null
        zookeeper.sync.time.ms = 2000
 (kafka.server.KafkaConfig)
[2020-05-27 04:09:28,930] INFO [ThrottledChannelReaper-Fetch]: Starting (kafka.server.ClientQuotaManager$ThrottledChannelReaper)
[2020-05-27 04:09:28,931] INFO [ThrottledChannelReaper-Produce]: Starting (kafka.server.ClientQuotaManager$ThrottledChannelReaper)
[2020-05-27 04:09:28,933] INFO [ThrottledChannelReaper-Request]: Starting (kafka.server.ClientQuotaManager$ThrottledChannelReaper)
[2020-05-27 04:09:28,972] INFO Log directory /tmp/kafka-logs not found, creating it. (kafka.log.LogManager)
[2020-05-27 04:09:28,982] INFO Loading logs. (kafka.log.LogManager)
[2020-05-27 04:09:29,002] INFO Logs loading complete in 20 ms. (kafka.log.LogManager)
[2020-05-27 04:09:29,026] INFO Starting log cleanup with a period of 300000 ms. (kafka.log.LogManager)
[2020-05-27 04:09:29,044] INFO Starting log flusher with a default period of 9223372036854775807 ms. (kafka.log.LogManager)
[2020-05-27 04:09:29,859] INFO Awaiting socket connections on localhost:9092. (kafka.network.Acceptor)
[2020-05-27 04:09:29,948] INFO [SocketServer brokerId=0] Created data-plane acceptor and processors for endpoint : EndPoint(localhost,9092,ListenerName(PLAINTEXT),PLAINTEXT) (kafka.network.SocketServer)
[2020-05-27 04:09:29,949] INFO [SocketServer brokerId=0] Started 1 acceptor threads for data-plane (kafka.network.SocketServer)
[2020-05-27 04:09:29,999] INFO [ExpirationReaper-0-Produce]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2020-05-27 04:09:30,000] INFO [ExpirationReaper-0-Fetch]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2020-05-27 04:09:30,007] INFO [ExpirationReaper-0-DeleteRecords]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2020-05-27 04:09:30,015] INFO [ExpirationReaper-0-ElectLeader]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2020-05-27 04:09:30,044] INFO [LogDirFailureHandler]: Starting (kafka.server.ReplicaManager$LogDirFailureHandler)
[2020-05-27 04:09:30,070] INFO Creating /brokers/ids/0 (is it secure? false) (kafka.zk.KafkaZkClient)
[2020-05-27 04:09:30,097] INFO Stat of the created znode at /brokers/ids/0 is: 24,24,1590552570089,1590552570089,1,0,0,72174634783080448,188,0,24
 (kafka.zk.KafkaZkClient)
[2020-05-27 04:09:30,099] INFO Registered broker 0 at path /brokers/ids/0 with addresses: ArrayBuffer(EndPoint(localhost,9092,ListenerName(PLAINTEXT),PLAINTEXT)), czxid (broker epoch): 24 (kafka.zk.KafkaZ
kClient)
[2020-05-27 04:09:30,215] INFO [ExpirationReaper-0-topic]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2020-05-27 04:09:30,226] INFO [ExpirationReaper-0-Heartbeat]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2020-05-27 04:09:30,237] INFO Successfully created /controller_epoch with initial epoch 0 (kafka.zk.KafkaZkClient)
[2020-05-27 04:09:30,243] INFO [ExpirationReaper-0-Rebalance]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2020-05-27 04:09:30,327] INFO [GroupCoordinator 0]: Starting up. (kafka.coordinator.group.GroupCoordinator)
[2020-05-27 04:09:30,332] INFO [GroupCoordinator 0]: Startup complete. (kafka.coordinator.group.GroupCoordinator)
[2020-05-27 04:09:30,350] INFO [GroupMetadataManager brokerId=0] Removed 0 expired offsets in 18 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-05-27 04:09:30,371] INFO [ProducerId Manager 0]: Acquired new producerId block (brokerId:0,blockStartProducerId:0,blockEndProducerId:999) by writing to Zk with path version 1 (kafka.coordinator.tran
saction.ProducerIdManager)
[2020-05-27 04:09:30,451] INFO [TransactionCoordinator id=0] Starting up. (kafka.coordinator.transaction.TransactionCoordinator)
[2020-05-27 04:09:30,460] INFO [TransactionCoordinator id=0] Startup complete. (kafka.coordinator.transaction.TransactionCoordinator)
[2020-05-27 04:09:30,463] INFO [Transaction Marker Channel Manager 0]: Starting (kafka.coordinator.transaction.TransactionMarkerChannelManager)
[2020-05-27 04:09:30,581] INFO [ExpirationReaper-0-AlterAcls]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2020-05-27 04:09:30,664] INFO [/config/changes-event-process-thread]: Starting (kafka.common.ZkNodeChangeNotificationListener$ChangeEventProcessThread)
[2020-05-27 04:09:30,711] INFO [SocketServer brokerId=0] Started data-plane processors for 1 acceptors (kafka.network.SocketServer)
[2020-05-27 04:09:30,719] INFO Kafka version: 2.5.0 (org.apache.kafka.common.utils.AppInfoParser)
[2020-05-27 04:09:30,720] INFO Kafka commitId: 66563e712b0b9f84 (org.apache.kafka.common.utils.AppInfoParser)
[2020-05-27 04:09:30,720] INFO Kafka startTimeMs: 1590552570712 (org.apache.kafka.common.utils.AppInfoParser)
[2020-05-27 04:09:30,724] INFO [KafkaServer id=0] started (kafka.server.KafkaServer)
```

Kafka broker with id -0, started on instance 1

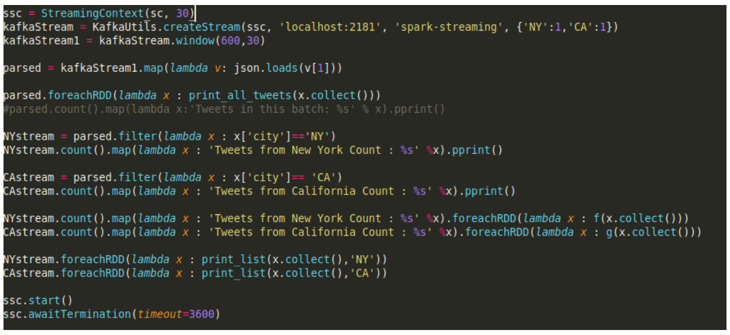


```
        zookeeper.ssl.cipher.suites = null
        zookeeper.ssl.client.enable = false
        zookeeper.ssl.crl.enable = false
        zookeeper.ssl.enabled.protocols = null
        zookeeper.ssl.endpoint.identification.algorithm = HTTPS
        zookeeper.ssl.keystore.location = null
        zookeeper.ssl.keystore.password = null
        zookeeper.ssl.keystore.type = null
        zookeeper.ssl.ocsp.enable = false
        zookeeper.ssl.protocol = TLSv1.2
        zookeeper.ssl.truststore.location = null
        zookeeper.ssl.truststore.password = null
        zookeeper.ssl.truststore.type = null
        zookeeper.sync.time.ms = 2000
(kafka.server.KafkaConfig)
[2020-05-27 04:10:26,326] INFO [ThrottledChannelReaper-Fetch]: Starting (kafka.server.ClientQuotaManager$ThrottledChannelReaper)
[2020-05-27 04:10:26,328] INFO [ThrottledChannelReaper-Produce]: Starting (kafka.server.ClientQuotaManager$ThrottledChannelReaper)
[2020-05-27 04:10:26,329] INFO [ThrottledChannelReaper-Request]: Starting (kafka.server.ClientQuotaManager$ThrottledChannelReaper)
[2020-05-27 04:10:26,374] INFO Log directory /tmp/kafka-logs not found, creating it. (kafka.log.LogManager)
[2020-05-27 04:10:26,386] INFO Loading logs. (kafka.log.LogManager)
[2020-05-27 04:10:26,409] INFO Logs loading complete in 21 ms. (kafka.log.LogManager)
[2020-05-27 04:10:26,436] INFO Starting log cleanup with a period of 300000 ms. (kafka.log.LogManager)
[2020-05-27 04:10:26,441] INFO Starting log flusher with a default period of 9223372036854775807 ms. (kafka.log.LogManager)
[2020-05-27 04:10:27,291] INFO Awaiting socket connections on 0.0.0.0:9092. (kafka.network.Acceptor)
[2020-05-27 04:10:27,387] INFO [SocketServer brokerId=1] Created data-plane acceptor and processors for endpoint : EndPoint(null,9092,ListenerName(PLAINTEXT),PLAINTEXT) (kafka.network.SocketServer)
[2020-05-27 04:10:27,388] INFO [SocketServer brokerId=1] Started 1 acceptor threads for data-plane (kafka.network.SocketServer)
[2020-05-27 04:10:27,417] INFO [ExpirationReaper-1-Produce]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2020-05-27 04:10:27,418] INFO [ExpirationReaper-1-Fetch]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2020-05-27 04:10:27,419] INFO [ExpirationReaper-1-DeleteRecords]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2020-05-27 04:10:27,424] INFO [ExpirationReaper-1-ElectLeader]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2020-05-27 04:10:27,450] INFO [LogDirFailureHandler]: Starting (kafka.server.ReplicaManager$LogDirFailureHandler)
[2020-05-27 04:10:27,536] INFO Creating /brokers/ids/1 (is it secure? false) (kafka.zk.KafkaZkClient)
[2020-05-27 04:10:27,569] INFO Stat of the created znode at /brokers/ids/1 is: 43,43,1590552627563,1590552627563,1,0,0,72174634783080449,228,0,43
(kafka.zk.KafkaZkClient)
[2020-05-27 04:10:27,570] INFO Registered broker 1 at path /brokers/ids/1 with addresses: ArrayBuffer(EndPoint(ip-172-31-47-221.ec2.internal,9092,ListenerName(PLAINTEXT),PLAINTEXT)), czxid (broker epoch):
43 (kafka.zk.KafkaZkClient)
[2020-05-27 04:10:27,719] INFO [ExpirationReaper-1-topic]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2020-05-27 04:10:27,731] INFO [ExpirationReaper-1-Heartbeat]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2020-05-27 04:10:27,733] INFO [ExpirationReaper-1-Rebalance]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2020-05-27 04:10:27,768] INFO [GroupCoordinator 1]: Starting up. (kafka.coordinator.group.GroupCoordinator)
[2020-05-27 04:10:27,770] INFO [GroupCoordinator 1]: Startup complete. (kafka.coordinator.group.GroupCoordinator)
[2020-05-27 04:10:27,788] INFO [GroupMetadataManager brokerId=1] Removed 0 expired offsets in 18 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-05-27 04:10:27,794] INFO [ProducerId Manager 1]: Acquired new producerId block (brokerId:1,blockStartProducerId:1000,blockEndProducerId:1999) by writing to Zk with path version 2 (kafka.coordinator.
ransaction.ProducerIdManager)
[2020-05-27 04:10:27,852] INFO [TransactionCoordinator id=1] Starting up. (kafka.coordinator.transaction.TransactionCoordinator)
[2020-05-27 04:10:27,855] INFO [TransactionCoordinator id=1] Startup complete. (kafka.coordinator.transaction.TransactionCoordinator)
[2020-05-27 04:10:27,861] INFO [Transaction Marker Channel Manager 1]: Starting (kafka.coordinator.transaction.TransactionMarkerChannelManager)
[2020-05-27 04:10:27,922] INFO [ExpirationReaper-1-AlterAcls]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2020-05-27 04:10:27,970] INFO [/config/changes-event-process-thread]: Starting (kafka.common.ZkNodeChangeNotificationListener$ChangeEventProcessThread)
[2020-05-27 04:10:28,002] INFO [SocketServer brokerId=1] Started data-plane processors for 1 acceptors (kafka.network.SocketServer)
[2020-05-27 04:10:28,011] INFO Kafka version: 2.5.0 (org.apache.kafka.common.utils.AppInfoParser)
[2020-05-27 04:10:28,011] INFO Kafka commitId: 66563e712b0b9f84 (org.apache.kafka.common.utils.AppInfoParser)
[2020-05-27 04:10:28,018] INFO Kafka startTimeMs: 1590552628003 (org.apache.kafka.common.utils.AppInfoParser)
[2020-05-27 04:10:28,021] INFO [KafkaServer id=1] started (kafka.server.KafkaServer)
```

Kafka broker with id - 1, started on instance 2

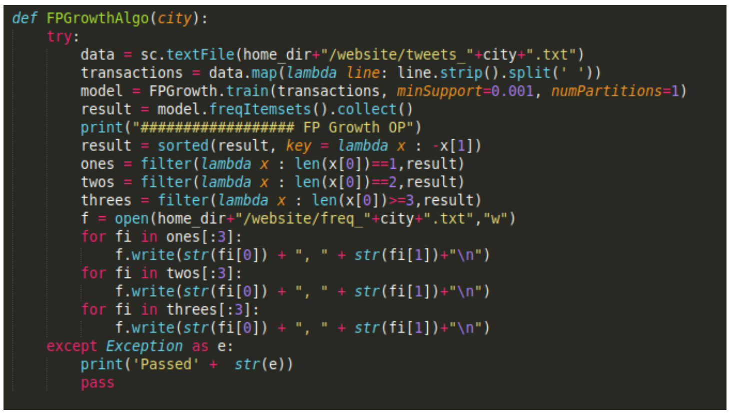
4 Spark Streaming

A stream is started, and tweets are pulled every 30 seconds. A window of 600 seconds is used, and all the tweets in this window are used for performing tweet count, and the frequent item set algorithm. The dstream cannot be used directly for the fp growth algorithm, and so, each rdd is written to a text file and this is used as input for the algorithm. The output of the algorithm is written to another text file, which is displayed in the website.



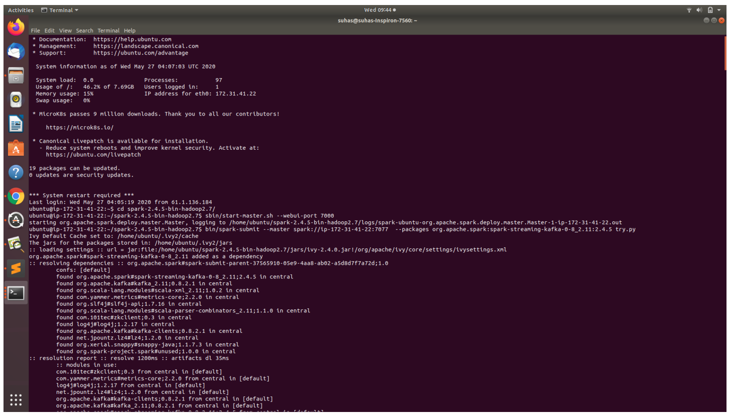
```
ssc = StreamingContext(sc, 30)
kafkaStream = KafkaUtils.createStream(ssc, 'localhost:2181', 'spark-streaming', {'NY':1,'CA':1})
kafkaStream1 = kafkaStream.window(600,30)

parsed = kafkaStream1.map(lambda v: json.loads(v[1]))

parsed.foreachRDD(lambda x : print_all_tweets(x.collect()))
#parsed.count().map(lambda x:'Tweets in this batch: %s' % x).pprint()

NYstream = parsed.filter(lambda x : x['city']=='NY')
NYstream.count().map(lambda x : 'Tweets from New York Count : %s' %x).pprint()

CAstream = parsed.filter(lambda x : x['city']== 'CA')
CAstream.count().map(lambda x : 'Tweets from California Count : %s' %x).pprint()

NYstream.count().map(lambda x : 'Tweets from New York Count : %s' %x).foreachRDD(lambda x : f(x.collect()))
CAstream.count().map(lambda x : 'Tweets from California Count : %s' %x).foreachRDD(lambda x : g(x.collect()))

NYstream.foreachRDD(lambda x : print_list(x.collect(),'NY'))
CAstream.foreachRDD(lambda x : print_list(x.collect(),'CA'))

ssc.start()
ssc.awaitTermination(timeout=3600)
```

Relevant code for the Spark stream



```
def FPGrowthAlgo(city):
    try:
        data = sc.textFile(home_dir+"/website/tweets_"+city+".txt")
        transactions = data.map(lambda line: line.strip().split(' '))
        model = FPGrowth.train(transactions, minSupport=0.001, numPartitions=1)
        result = model.freqItemsets().collect()
        print("################# FP Growth OP")
        result = sorted(result, key = lambda x : -x[1])
        ones = filter(lambda x : len(x[0])==1,result)
        twos = filter(lambda x : len(x[0])==2,result)
        threes = filter(lambda x : len(x[0])>=3,result)
        f = open(home_dir+"/website/freq_"+city+".txt","w")
        for fi in ones[:3]:
            f.write(str(fi[0]) + ", " + str(fi[1])+"\n")
        for fi in twos[:3]:
            f.write(str(fi[0]) + ", " + str(fi[1])+"\n")
        for fi in threes[:3]:
            f.write(str(fi[0]) + ", " + str(fi[1])+"\n")
    except Exception as e:
        print('Passed' +  str(e))
        pass
```

Fp Growth algorithm



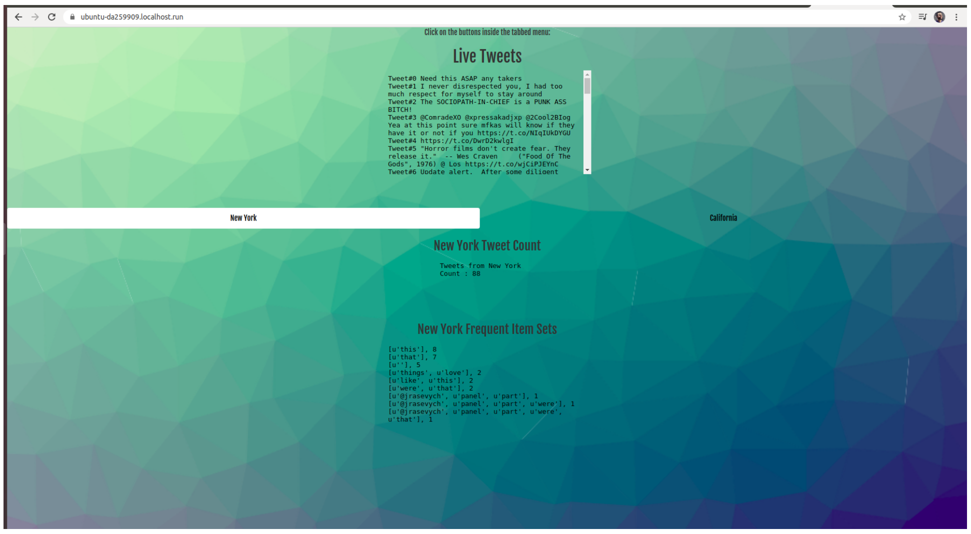
Activities
Terminal
Wed 09:44
suhas@suhas-Inspiron-7560: ~
File Edit View Search Terminal Help

```
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed May 27 04:07:03 UTC 2020

  System load:  0.0               Processes:           97
  Usage of /:   46.2% of 7.69GB   Users logged in:     1
  Memory usage: 15%               IP address for eth0: 172.31.41.22
  Swap usage:   0%

 * MicroK8s passes 9 million downloads. Thank you to all our contributors!

     https://microk8s.io/

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

19 packages can be updated.
0 updates are security updates.


*** System restart required ***
Last login: Wed May 27 04:05:19 2020 from 61.1.136.184
ubuntu@ip-172-31-41-22:~$ cd spark-2.4.5-bin-hadoop2.7/
ubuntu@ip-172-31-41-22:~/spark-2.4.5-bin-hadoop2.7$ sbin/start-master.sh --webui-port 7000
starting org.apache.spark.deploy.master.Master, logging to /home/ubuntu/spark-2.4.5-bin-hadoop2.7/logs/spark-ubuntu-org.apache.spark.deploy.master.Master-1-ip-172-31-41-22.out
ubuntu@ip-172-31-41-22:~/spark-2.4.5-bin-hadoop2.7$ bin/spark-submit --master spark://ip-172-31-41-22:7077  --packages org.apache.spark:spark-streaming-kafka-0-8_2.11:2.4.5 try.py
Ivy Default Cache set to: /home/ubuntu/.ivy2/cache
The jars for the packages stored in: /home/ubuntu/.ivy2/jars
:: loading settings :: url = jar:file:/home/ubuntu/spark-2.4.5-bin-hadoop2.7/jars/ivy-2.4.0.jar!/org/apache/ivy/core/settings/ivysettings.xml
org.apache.spark#spark-streaming-kafka-0-8_2.11 added as a dependency
:: resolving dependencies :: org.apache.spark#spark-submit-parent-37565910-05e9-4aa8-ab02-a5d8d7f7a72d;1.0
        confs: [default]
        found org.apache.spark#spark-streaming-kafka-0-8_2.11;2.4.5 in central
        found org.apache.kafka#kafka_2.11;0.8.2.1 in central
        found org.scala-lang.modules#scala-xml_2.11;1.0.2 in central
        found com.yammer.metrics#metrics-core;2.2.0 in central
        found org.slf4j#slf4j-api;1.7.16 in central
        found org.scala-lang.modules#scala-parser-combinators_2.11;1.1.0 in central
        found com.101tec#zkclient;0.3 in central
        found log4j#log4j;1.2.17 in central
        found org.apache.kafka#kafka-clients;0.8.2.1 in central
        found net.jpountz.lz4#lz4;1.2.0 in central
        found org.xerial.snappy#snappy-java;1.1.7.3 in central
        found org.spark-project.spark#unused;1.0.0 in central
:: resolution report :: resolve 1200ms :: artifacts dl 35ms
        :: modules in use:
        com.101tec#zkclient;0.3 from central in [default]
        com.yammer.metrics#metrics-core;2.2.0 from central in [default]
        log4j#log4j;1.2.17 from central in [default]
        net.jpountz.lz4#lz4;1.2.0 from central in [default]
        org.apache.kafka#kafka-clients;0.8.2.1 from central in [default]
        org.apache.kafka#kafka_2.11;0.8.2.1 from central in [default]
```

The command for starting spark, and submitting the job

5 Relevant Screenshots

The website reloads every 15 seconds, and outputs the data of the Spark Fp Growth algorithm. More proof of working can be found in the video attached along with this report.

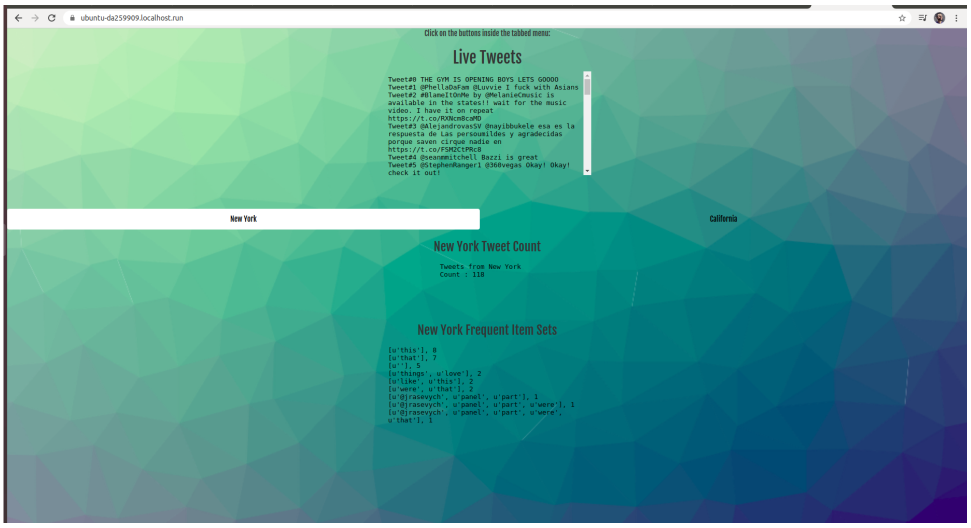

ubuntu-da259909.localhost.run
Click on the buttons inside the tabbed menu:
Live Tweets
New York
California
New York Tweet Count
New York Frequent Item Sets

```
Tweet#0 Need this ASAP any takers
Tweet#1 I never disrespected you, I had too
much respect for myself to stay around
Tweet#2 The SOCIOPATH-IN-CHIEF is a PUNK ASS
BITCH!
Tweet#3 @ComradeXO @xpressakadjxp @2Cool2BIog
Yea at this point sure mfkas will know if they
have it or not if you https://t.co/NIqIUkDYGU
Tweet#4 https://t.co/DwrD2kwlgI
Tweet#5 "Horror films don't create fear. They
release it."  -- Wes Craven     ("Food Of The
Gods", 1976) @ Los https://t.co/wjCiPJEYnC
Tweet#6 Update alert.  After some diligent
```

```
Tweets from New York
Count : 88
```

```
[u'this'], 8
[u'that'], 7
[u''], 5
[u'things', u'love'], 2
[u'like', u'this'], 2
[u'were', u'that'], 2
[u'@jrasevych', u'panel', u'part'], 1
[u'@jrasevych', u'panel', u'part', u'were'], 1
[u'@jrasevych', u'panel', u'part', u'were',
u'that'], 1
```

Initial New York Tweet count


ubuntu-da259909.localhost.run
Click on the buttons inside the tabbed menu:
Live Tweets
New York
California
New York Tweet Count
New York Frequent Item Sets

```
Tweet#0 THE GYM IS OPENING BOYS LETS GOOOO
Tweet#1 @PhellaDaFam @Luvvie I fuck with Asians
Tweet#2 #BlameItOnMe by @MelanieCmusic is
available in the states!! wait for the music
video. I have it on repeat
https://t.co/RXNcm8caMD
Tweet#3 @AlejandrovasSV @nayibbukele esa es la
respuesta de Las persoumildes y agradecidas
porque saven cirque nadie en
https://t.co/FSM2CtPRc8
Tweet#4 @seanmmitchell Bazzi is great
Tweet#5 @StephenRanger1 @360vegas Okay! Okay!
check it out!
```

```
Tweets from New York
Count : 118
```

```
[u'this'], 8
[u'that'], 7
[u''], 5
[u'things', u'love'], 2
[u'like', u'this'], 2
[u'were', u'that'], 2
[u'@jrasevych', u'panel', u'part'], 1
[u'@jrasevych', u'panel', u'part', u'were'], 1
[u'@jrasevych', u'panel', u'part', u'were',
u'that'], 1
```

New York Tweet count after 30 seconds