IERG 4330/ ESTR4316 /IEMS 5730 Spring 2022 Homework 4

Release date: Mar 31, 2022

Due date: 23:59:00, Apr 19, 2022

The solution will be posted right after the deadline, so no late homework will be accepted!

Every Student MUST include the following statement, together with his/her signature in the submitted homework.

I declare that the assignment submitted on Blackboard system is original except for source material explicitly acknowledged, and that thea same or related material has not been previously submitted for another course. I also acknowledge that I am aware of University policy and regulations on honesty in academic work, and of the disciplinary guidelines and procedures applicable to breaches of such policy and regulations, as contained in the website http://www.cuhk.edu.hk/policy/academichonesty/.

Signed (Student Eddie Christopher Fox III) Date: April 18, 2022

Name Eddie Christopher Fox III SID 1155160788

Submission notice:

- Submit your homework via the elearning system

General homework policies:

A student may discuss the problems with others. However, the work a student turns in must be created COMPLETELY by oneself ALONE. A student may not share ANY written work or pictures, nor may one copy answers from any source other than one's own brain.

Each student **MUST LIST** on the homework paper the **name of every person he/she has discussed or worked with**. If the answer includes content from any other source, the student **MUST STATE THE SOURCE**. Failure to do so is cheating and will result in sanctions. Copying answers from someone else is cheating even if one lists their name(s) on the homework.

If there is information you need to solve a problem but the information is not stated in the problem, try to find the data somewhere. If you cannot find it, state what data you need, make a reasonable estimate of its value and justify any assumptions you make. You will be graded not only on whether your answer is correct but also on whether you have done an intelligent analysis.

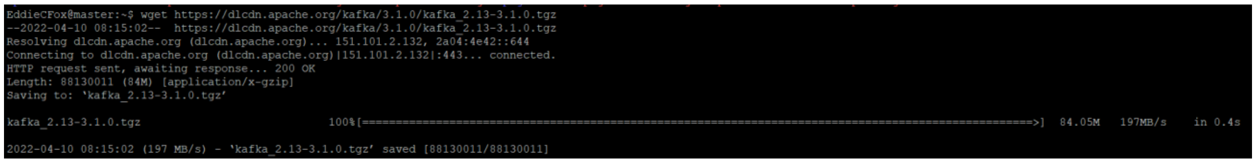
Eddie Christopher Fox III
Student ID: 1155160788

IEMS 5730: Spring 2022 Assignment #4

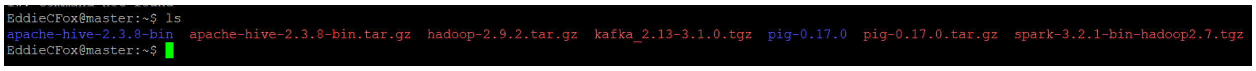
Question 1 Section A [Bonus]: Multi-node Kafka Cluster Setup

Most commands are shown from the master node unless there is notable differences on a slave node.

I first download Kafka onto each node from [1] as suggested in [2] by running "wget https://dlcdn.apache.org/kafka/3.1.0/kafka_2.13-3.1.0.tgz".

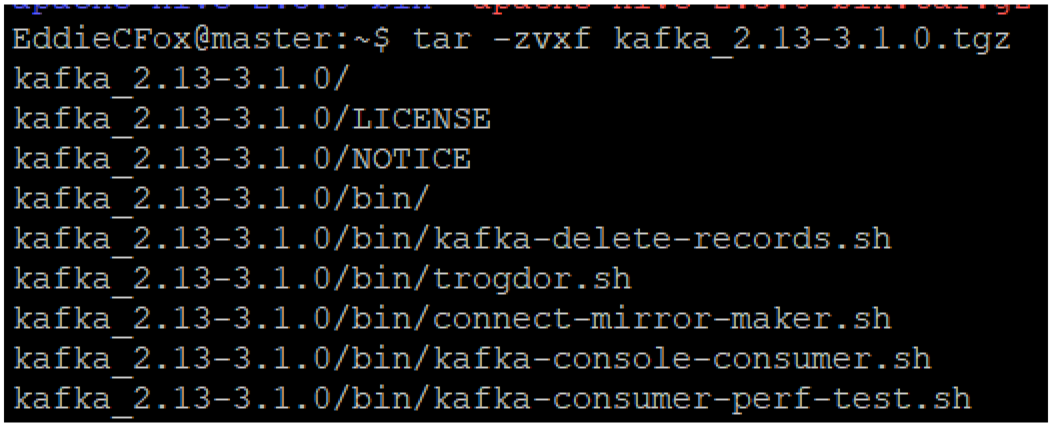


```
EddieCFox@master:~$ wget https://dlcdn.apache.org/kafka/3.1.0/kafka_2.13-3.1.0.tgz
--2022-04-10 08:15:02--  https://dlcdn.apache.org/kafka/3.1.0/kafka_2.13-3.1.0.tgz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 88130011 (84M) [application/x-gzip]
Saving to: 'kafka_2.13-3.1.0.tgz'

kafka_2.13-3.1.0.tgz                    100%[==========================================================================================>]  84.05M   197MB/s    in 0.4s

2022-04-10 08:15:02 (197 MB/s) - 'kafka_2.13-3.1.0.tgz' saved [88130011/88130011]
```



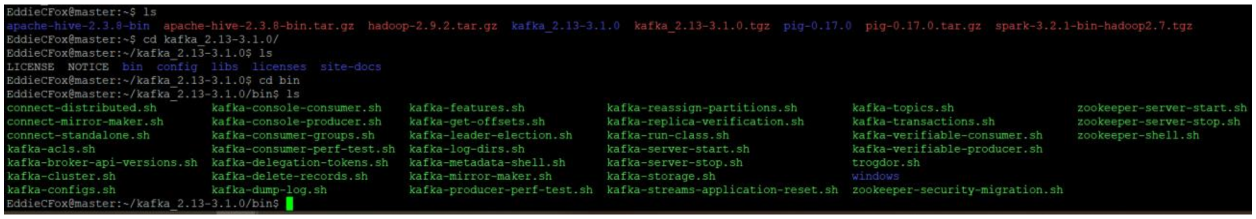
```
EddieCFox@master:~$ ls
apache-hive-2.3.8-bin  apache-hive-2.3.8-bin.tar.gz  hadoop-2.9.2.tar.gz  kafka_2.13-3.1.0.tgz  pig-0.17.0  pig-0.17.0.tar.gz  spark-3.2.1-bin-hadoop2.7.tgz
EddieCFox@master:~$
```

I then run "tar -zvxf kafka_2.13-3.1.0.tgz" on each node.



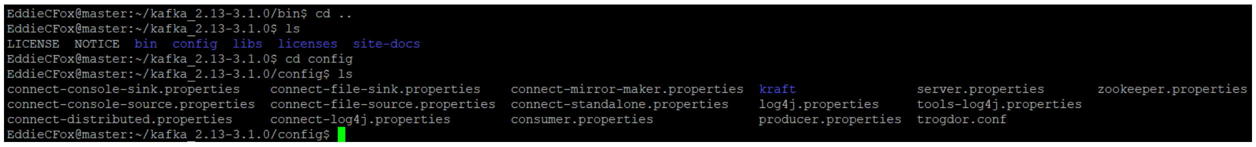
```
EddieCFox@master:~$ tar -zvxf kafka_2.13-3.1.0.tgz
kafka_2.13-3.1.0/
kafka_2.13-3.1.0/LICENSE
kafka_2.13-3.1.0/NOTICE
kafka_2.13-3.1.0/bin/
kafka_2.13-3.1.0/bin/kafka-delete-records.sh
kafka_2.13-3.1.0/bin/trogdor.sh
kafka_2.13-3.1.0/bin/connect-mirror-maker.sh
kafka_2.13-3.1.0/bin/kafka-console-consumer.sh
kafka_2.13-3.1.0/bin/kafka-consumer-perf-test.sh
```

Output omitted.



```
EddieCFox@master:~$ ls
apache-hive-2.3.8-bin  apache-hive-2.3.8-bin.tar.gz  hadoop-2.9.2.tar.gz  kafka_2.13-3.1.0  kafka_2.13-3.1.0.tgz  pig-0.17.0  pig-0.17.0.tar.gz  spark-3.2.1-bin-hadoop2.7.tgz
EddieCFox@master:~$ cd kafka_2.13-3.1.0/
EddieCFox@master:~/kafka_2.13-3.1.0$ ls
LICENSE  NOTICE  bin  config  libs  licenses  site-docs
EddieCFox@master:~/kafka_2.13-3.1.0$ cd bin
EddieCFox@master:~/kafka_2.13-3.1.0/bin$ ls
connect-distributed.sh        kafka-console-consumer.sh    kafka-features.sh             kafka-reassign-partitions.sh        kafka-topics.sh                  zookeeper-server-start.sh
connect-mirror-maker.sh       kafka-console-producer.sh    kafka-get-offsets.sh          kafka-replica-verification.sh       kafka-transactions.sh            zookeeper-server-stop.sh
connect-standalone.sh         kafka-consumer-groups.sh     kafka-leader-election.sh      kafka-run-class.sh                  kafka-verifiable-consumer.sh     zookeeper-shell.sh
kafka-acls.sh                 kafka-consumer-perf-test.sh  kafka-log-dirs.sh             kafka-server-start.sh               kafka-verifiable-producer.sh
kafka-broker-api-versions.sh  kafka-delegation-tokens.sh   kafka-metadata-shell.sh       kafka-server-stop.sh                trogdor.sh
kafka-cluster.sh              kafka-delete-records.sh      kafka-mirror-maker.sh         kafka-storage.sh                    windows
kafka-configs.sh              kafka-dump-log.sh            kafka-producer-perf-test.sh   kafka-streams-application-reset.sh  zookeeper-security-migration.sh
EddieCFox@master:~/kafka_2.13-3.1.0/bin$
```

I navigate into the bin folder of Kafka after extracting the Kafka gzip.

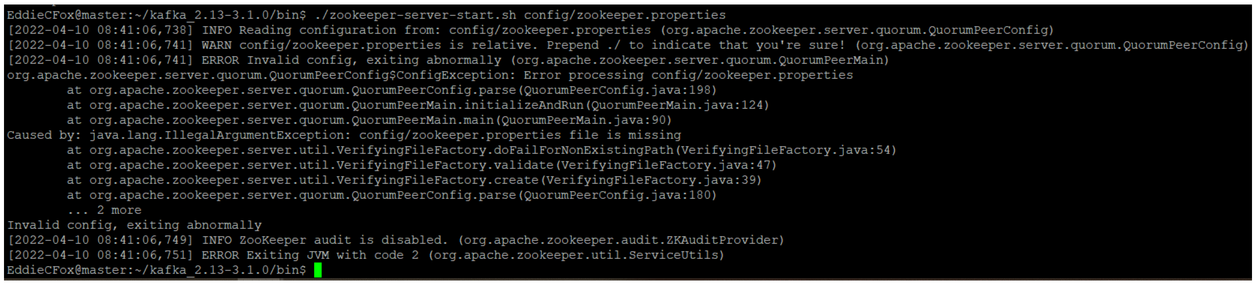


```
EddieCFox@master:~/kafka_2.13-3.1.0/bin$ cd ..
EddieCFox@master:~/kafka_2.13-3.1.0$ ls
LICENSE  NOTICE  bin  config  libs  licenses  site-docs
EddieCFox@master:~/kafka_2.13-3.1.0$ cd config
EddieCFox@master:~/kafka_2.13-3.1.0/config$ ls
connect-console-sink.properties    connect-file-sink.properties    connect-mirror-maker.properties  kraft                 server.properties        zookeeper.properties
connect-console-source.properties  connect-file-source.properties  connect-standalone.properties    log4j.properties      tools-log4j.properties
connect-distributed.properties     connect-log4j.properties        consumer.properties              producer.properties   trogdor.conf
EddieCFox@master:~/kafka_2.13-3.1.0/config$
```

The config folder has the zookeeper.properties and server.properties that we want to ensure are configured appropriately before starting the Zookeeper and Kafka servers[3].

Zookeeper defaults appear fine. The server.properties file contains the parameters mentioned under [4] by which we can use to, among other things, configure the number of partitions. It appears that the number of partitions is set when creating the target, but to be safe, I look at every property involving partitions and set the value to 2. After searching through [4] in Section 3.1: Broker Configs, these would be offsets.topic.num.partitions, transaction.state.log.num.partitions, and num.partitions.

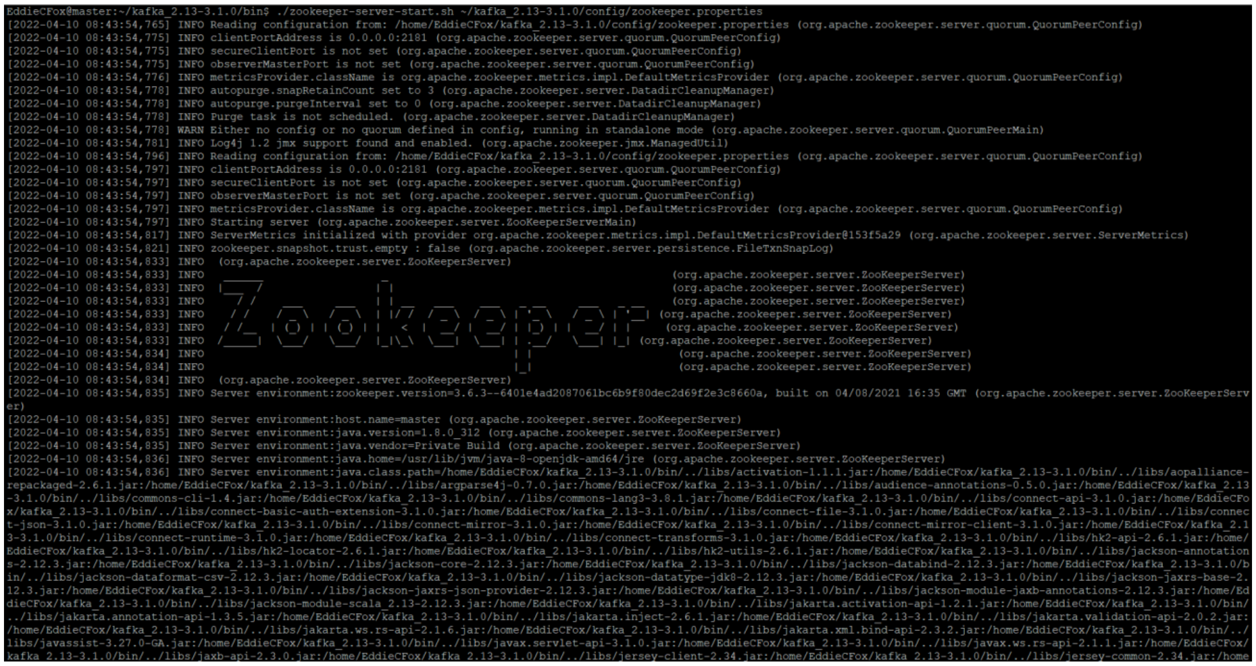
For this assignment, we need to create 2 brokers with 2 partitions. It seems that each broker will be on one machine, and I have 4 VM's total in my cluster, so I will run Zookeeper on the master node[3], then run broker1 on slave1 and broker2 on slave2.



```
EddieCFox@master:~/kafka_2.13-3.1.0/bin$ ./zookeeper-server-start.sh config/zookeeper.properties
[2022-04-10 08:41:06,738] INFO Reading configuration from: config/zookeeper.properties (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2022-04-10 08:41:06,741] WARN config/zookeeper.properties is relative. Prepend ./ to indicate that you're sure! (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2022-04-10 08:41:06,741] ERROR Invalid config, exiting abnormally (org.apache.zookeeper.server.quorum.QuorumPeerMain)
org.apache.zookeeper.server.quorum.QuorumPeerConfig$ConfigException: Error processing config/zookeeper.properties
        at org.apache.zookeeper.server.quorum.QuorumPeerConfig.parse(QuorumPeerConfig.java:198)
        at org.apache.zookeeper.server.quorum.QuorumPeerMain.initializeAndRun(QuorumPeerMain.java:124)
        at org.apache.zookeeper.server.quorum.QuorumPeerMain.main(QuorumPeerMain.java:90)
Caused by: java.lang.IllegalArgumentException: config/zookeeper.properties file is missing
        at org.apache.zookeeper.server.util.VerifyingFileFactory.doFailForNonExistingPath(VerifyingFileFactory.java:54)
        at org.apache.zookeeper.server.util.VerifyingFileFactory.validate(VerifyingFileFactory.java:47)
        at org.apache.zookeeper.server.util.VerifyingFileFactory.create(VerifyingFileFactory.java:39)
        at org.apache.zookeeper.server.quorum.QuorumPeerConfig.parse(QuorumPeerConfig.java:180)
        ... 2 more
Invalid config, exiting abnormally
[2022-04-10 08:41:06,749] INFO ZooKeeper audit is disabled. (org.apache.zookeeper.audit.ZKAuditProvider)
[2022-04-10 08:41:06,751] ERROR Exiting JVM with code 2 (org.apache.zookeeper.util.ServiceUtils)
EddieCFox@master:~/kafka_2.13-3.1.0/bin$ 
```

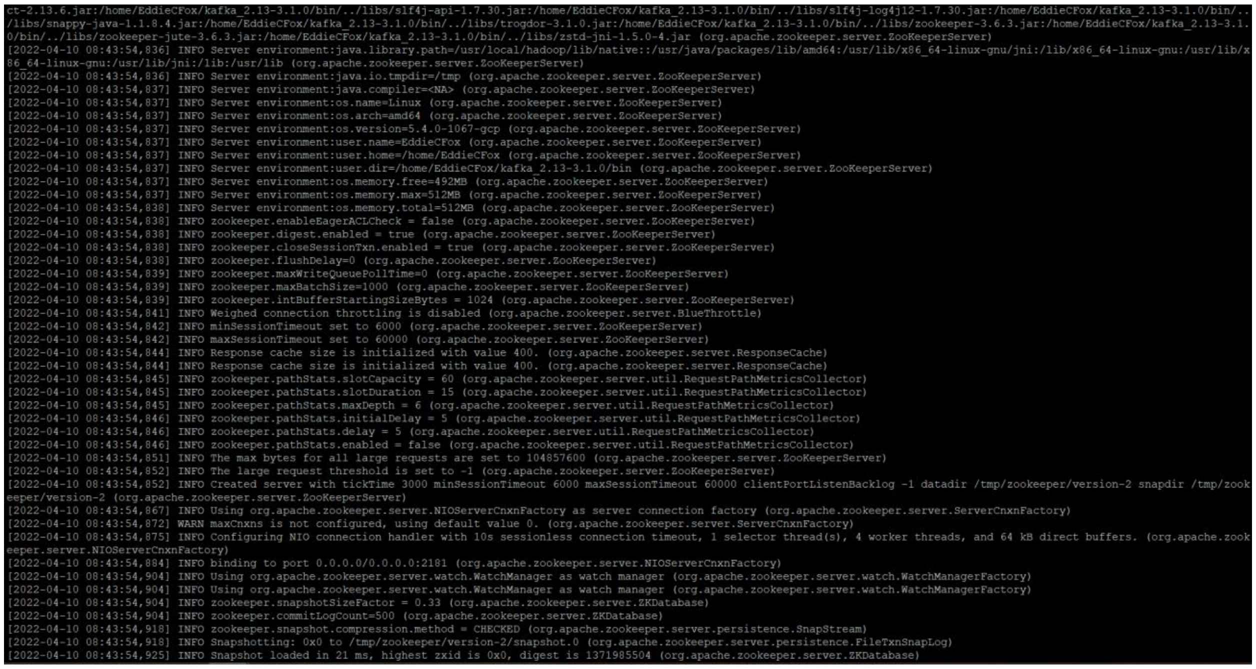
I try running zookeeper on the master node using the command in [2] but receive this error. I think I need to explicitly give the directory where zookeeper properties is.

After running the following command, it works: "./zookeeper-server-start.sh ~/kafka_2.13-3.1.0/config/zookeeper.properties"

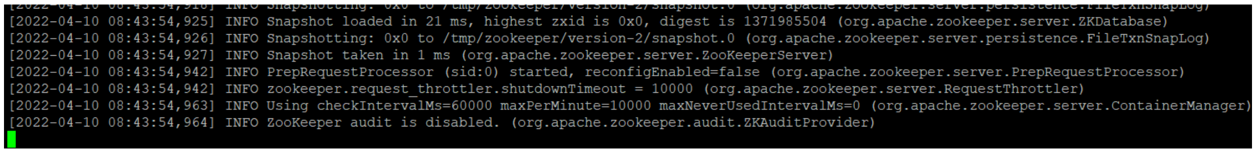


```
EddieCFox@master:~/kafka_2.13-3.1.0/bin$ ./zookeeper-server-start.sh ~/kafka_2.13-3.1.0/config/zookeeper.properties
[2022-04-10 08:43:54,765] INFO Reading configuration from: /home/EddieCFox/kafka_2.13-3.1.0/config/zookeeper.properties (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2022-04-10 08:43:54,775] INFO clientPortAddress is 0.0.0.0:2181 (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2022-04-10 08:43:54,775] INFO secureClientPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2022-04-10 08:43:54,775] INFO observerMasterPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2022-04-10 08:43:54,776] INFO metricsProvider.className is org.apache.zookeeper.metrics.impl.DefaultMetricsProvider (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2022-04-10 08:43:54,778] INFO autopurge.snapRetainCount set to 3 (org.apache.zookeeper.server.DatadirCleanupManager)
[2022-04-10 08:43:54,778] INFO autopurge.purgeInterval set to 0 (org.apache.zookeeper.server.DatadirCleanupManager)
[2022-04-10 08:43:54,778] INFO Purge task is not scheduled. (org.apache.zookeeper.server.DatadirCleanupManager)
[2022-04-10 08:43:54,778] WARN Either no config or no quorum defined in config, running in standalone mode (org.apache.zookeeper.server.quorum.QuorumPeerMain)
[2022-04-10 08:43:54,781] INFO Log4j 1.2 jmx support found and enabled. (org.apache.zookeeper.jmx.ManagedUtil)
[2022-04-10 08:43:54,796] INFO Reading configuration from: /home/EddieCFox/kafka_2.13-3.1.0/config/zookeeper.properties (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2022-04-10 08:43:54,797] INFO clientPortAddress is 0.0.0.0:2181 (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2022-04-10 08:43:54,797] INFO secureClientPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2022-04-10 08:43:54,797] INFO observerMasterPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2022-04-10 08:43:54,797] INFO metricsProvider.className is org.apache.zookeeper.metrics.impl.DefaultMetricsProvider (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2022-04-10 08:43:54,797] INFO Starting server (org.apache.zookeeper.server.ZooKeeperServerMain)
[2022-04-10 08:43:54,817] INFO ServerMetrics initialized with provider org.apache.zookeeper.metrics.impl.DefaultMetricsProvider@153f5a29 (org.apache.zookeeper.server.ServerMetrics)
[2022-04-10 08:43:54,821] INFO zookeeper.snapshot.trust.empty : false (org.apache.zookeeper.server.persistence.FileTxnSnapLog)
[2022-04-10 08:43:54,833] INFO  (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,833] INFO   ______                  _                                           (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,833] INFO  |___  /                 | |                                          (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,833] INFO     / /    ___     ___   | | __   ___    ___   _ __     ___   _ __    (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,833] INFO    / /    / _ \   / _ \  | |/ /  / _ \  / _ \ | '_ \   / _ \ | '__| (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,833] INFO   / /__  | (_) | | (_) | |   <  |  __/ |  __/ | |_) | |  __/ | |     (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,833] INFO  /_____|  \___/   \___/  |_|\_\  \___|  \___| | .__/   \___| |_| (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,834] INFO                                               | |                      (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,834] INFO                                               |_|                      (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,834] INFO  (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,835] INFO Server environment:zookeeper.version=3.6.3--6401e4ad2087061bc6b9f80dec2d69f2e3c8660a, built on 04/08/2021 16:35 GMT (org.apache.zookeeper.server.ZooKeeperServ
er)
[2022-04-10 08:43:54,835] INFO Server environment:host.name=master (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,835] INFO Server environment:java.version=1.8.0_312 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,835] INFO Server environment:java.vendor=Private Build (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,836] INFO Server environment:java.home=/usr/lib/jvm/java-8-openjdk-amd64/jre (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,836] INFO Server environment:java.class.path=/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/activation-1.1.1.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/aopalliance-
repackaged-2.6.1.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/argparse4j-0.7.0.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/audience-annotations-0.5.0.jar:/home/EddieCFox/kafka_2.13
-3.1.0/bin/../libs/commons-cli-1.4.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/commons-lang3-3.8.1.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/connect-api-3.1.0.jar:/home/EddieCFo
x/kafka_2.13-3.1.0/bin/../libs/connect-basic-auth-extension-3.1.0.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/connect-file-3.1.0.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/connec
t-json-3.1.0.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/connect-mirror-3.1.0.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/connect-mirror-client-3.1.0.jar:/home/EddieCFox/kafka_2.1
3-3.1.0/bin/../libs/connect-runtime-3.1.0.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/connect-transforms-3.1.0.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/hk2-api-2.6.1.jar:/home/
EddieCFox/kafka_2.13-3.1.0/bin/../libs/hk2-locator-2.6.1.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/hk2-utils-2.6.1.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/jackson-annotation
s-2.12.3.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/jackson-core-2.12.3.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/jackson-databind-2.12.3.jar:/home/EddieCFox/kafka_2.13-3.1.0/b
in/../libs/jackson-dataformat-csv-2.12.3.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/jackson-datatype-jdk8-2.12.3.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/jackson-jaxrs-base-2.
12.3.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/jackson-jaxrs-json-provider-2.12.3.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/jackson-module-jaxb-annotations-2.12.3.jar:/home/Ed
dieCFox/kafka_2.13-3.1.0/bin/../libs/jackson-module-scala_2.13-2.12.3.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/jakarta.activation-api-1.2.1.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/
../libs/jakarta.annotation-api-1.3.5.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/jakarta.inject-2.6.1.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/jakarta.validation-api-2.0.2.jar:
/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/jakarta.ws.rs-api-2.1.6.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/jakarta.xml.bind-api-2.3.2.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../
libs/javassist-3.27.0-GA.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/javax.servlet-api-3.1.0.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/javax.ws.rs-api-2.1.1.jar:/home/EddieCFox/
kafka_2.13-3.1.0/bin/../libs/jaxb-api-2.3.0.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/jersey-client-2.34.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/jersey-common-2.34.jar:/home
```

Some output omitted.

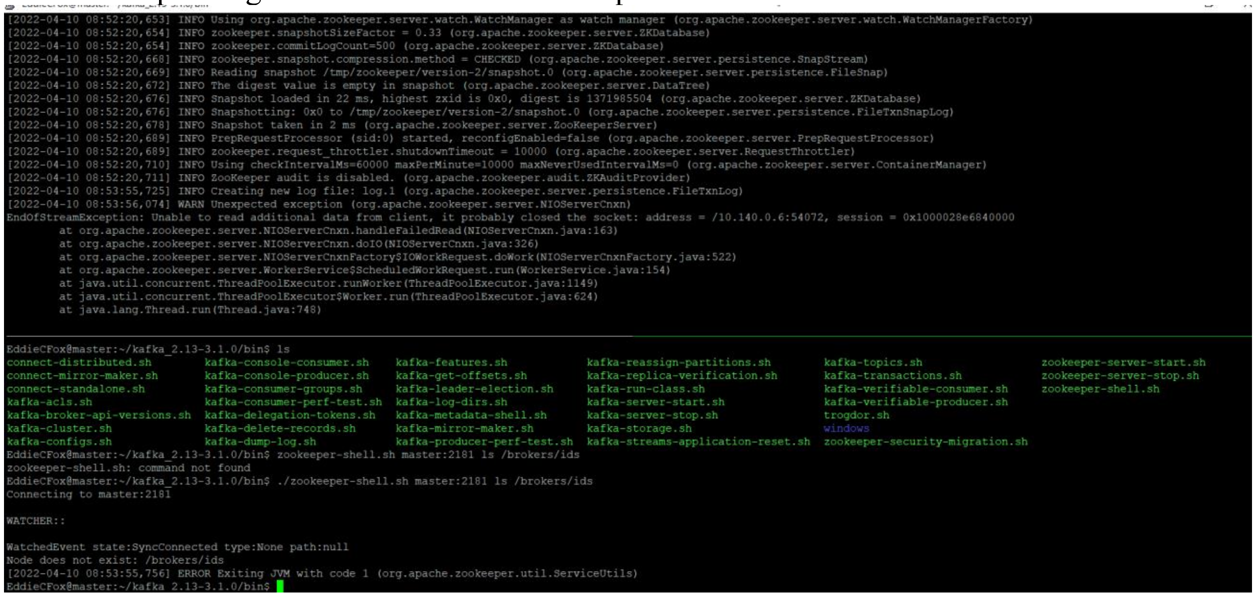


```
ct-2.13.6.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/slf4j-api-1.7.30.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/slf4j-log4j12-1.7.30.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/..
/libs/snappy-java-1.1.8.4.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/trogdor-3.1.0.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/zookeeper-3.6.3.jar:/home/EddieCFox/kafka_2.13-3.1.
0/bin/../libs/zookeeper-jute-3.6.3.jar:/home/EddieCFox/kafka_2.13-3.1.0/bin/../libs/zstd-jni-1.5.0-4.jar (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,836] INFO Server environment:java.library.path=/usr/local/hadoop/lib/native::/usr/java/packages/lib/amd64:/usr/lib/x86_64-linux-gnu/jni:/lib/x86_64-linux-gnu:/usr/lib/x
86_64-linux-gnu:/usr/lib/jni:/lib:/usr/lib (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,836] INFO Server environment:java.io.tmpdir=/tmp (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,837] INFO Server environment:java.compiler=<NA> (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,837] INFO Server environment:os.name=Linux (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,837] INFO Server environment:os.arch=amd64 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,837] INFO Server environment:os.version=5.4.0-1067-gcp (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,837] INFO Server environment:user.name=EddieCFox (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,837] INFO Server environment:user.home=/home/EddieCFox (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,837] INFO Server environment:user.dir=/home/EddieCFox/kafka_2.13-3.1.0/bin (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,837] INFO Server environment:os.memory.free=492MB (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,837] INFO Server environment:os.memory.max=512MB (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,838] INFO Server environment:os.memory.total=512MB (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,838] INFO zookeeper.enableEagerACLCheck = false (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,838] INFO zookeeper.digest.enabled = true (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,838] INFO zookeeper.closeSessionTxn.enabled = true (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,838] INFO zookeeper.flushDelay=0 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,839] INFO zookeeper.maxWriteQueuePollTime=0 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,839] INFO zookeeper.maxBatchSize=1000 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,839] INFO zookeeper.intBufferStartingSizeBytes = 1024 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,841] INFO Weighed connection throttling is disabled (org.apache.zookeeper.server.BlueThrottle)
[2022-04-10 08:43:54,842] INFO minSessionTimeout set to 6000 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,842] INFO maxSessionTimeout set to 60000 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,844] INFO Response cache size is initialized with value 400. (org.apache.zookeeper.server.ResponseCache)
[2022-04-10 08:43:54,844] INFO Response cache size is initialized with value 400. (org.apache.zookeeper.server.ResponseCache)
[2022-04-10 08:43:54,845] INFO zookeeper.pathStats.slotCapacity = 60 (org.apache.zookeeper.server.util.RequestPathMetricsCollector)
[2022-04-10 08:43:54,845] INFO zookeeper.pathStats.slotDuration = 15 (org.apache.zookeeper.server.util.RequestPathMetricsCollector)
[2022-04-10 08:43:54,845] INFO zookeeper.pathStats.maxDepth = 6 (org.apache.zookeeper.server.util.RequestPathMetricsCollector)
[2022-04-10 08:43:54,846] INFO zookeeper.pathStats.initialDelay = 5 (org.apache.zookeeper.server.util.RequestPathMetricsCollector)
[2022-04-10 08:43:54,846] INFO zookeeper.pathStats.delay = 5 (org.apache.zookeeper.server.util.RequestPathMetricsCollector)
[2022-04-10 08:43:54,846] INFO zookeeper.pathStats.enabled = false (org.apache.zookeeper.server.util.RequestPathMetricsCollector)
[2022-04-10 08:43:54,851] INFO The max bytes for all large requests are set to 104857600 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,852] INFO The large request threshold is set to -1 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,852] INFO Created server with tickTime 3000 minSessionTimeout 6000 maxSessionTimeout 60000 clientPortListenBacklog -1 datadir /tmp/zookeeper/version-2 snapdir /tmp/zook
eeper/version-2 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,867] INFO Using org.apache.zookeeper.server.NIOServerCnxnFactory as server connection factory (org.apache.zookeeper.server.ServerCnxnFactory)
[2022-04-10 08:43:54,872] WARN maxCnxns is not configured, using default value 0. (org.apache.zookeeper.server.ServerCnxnFactory)
[2022-04-10 08:43:54,875] INFO Configuring NIO connection handler with 10s sessionless connection timeout, 1 selector thread(s), 4 worker threads, and 64 kB direct buffers. (org.apache.zook
eeper.server.NIOServerCnxnFactory)
[2022-04-10 08:43:54,884] INFO binding to port 0.0.0.0/0.0.0.0:2181 (org.apache.zookeeper.server.NIOServerCnxnFactory)
[2022-04-10 08:43:54,904] INFO Using org.apache.zookeeper.server.watch.WatchManager as watch manager (org.apache.zookeeper.server.watch.WatchManagerFactory)
[2022-04-10 08:43:54,904] INFO Using org.apache.zookeeper.server.watch.WatchManager as watch manager (org.apache.zookeeper.server.watch.WatchManagerFactory)
[2022-04-10 08:43:54,904] INFO zookeeper.snapshotSizeFactor = 0.33 (org.apache.zookeeper.server.ZKDatabase)
[2022-04-10 08:43:54,904] INFO zookeeper.commitLogCount=500 (org.apache.zookeeper.server.ZKDatabase)
[2022-04-10 08:43:54,918] INFO zookeeper.snapshot.compression.method = CHECKED (org.apache.zookeeper.server.persistence.SnapStream)
[2022-04-10 08:43:54,918] INFO Snapshotting: 0x0 to /tmp/zookeeper/version-2/snapshot.0 (org.apache.zookeeper.server.persistence.FileTxnSnapLog)
[2022-04-10 08:43:54,925] INFO Snapshot loaded in 21 ms, highest zxid is 0x0, digest is 1371985504 (org.apache.zookeeper.server.ZKDatabase)
```



```
[2022-04-10 08:43:54,925] INFO Snapshot loaded in 21 ms, highest zxid is 0x0, digest is 1371985504 (org.apache.zookeeper.server.ZKDatabase)
[2022-04-10 08:43:54,926] INFO Snapshotting: 0x0 to /tmp/zookeeper/version-2/snapshot.0 (org.apache.zookeeper.server.persistence.FileTxnSnapLog)
[2022-04-10 08:43:54,927] INFO Snapshot taken in 1 ms (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:43:54,942] INFO PrepRequestProcessor (sid:0) started, reconfigEnabled=false (org.apache.zookeeper.server.PrepRequestProcessor)
[2022-04-10 08:43:54,942] INFO zookeeper.request_throttler.shutdownTimeout = 10000 (org.apache.zookeeper.server.RequestThrottler)
[2022-04-10 08:43:54,963] INFO Using checkIntervalMs=60000 maxPerMinute=10000 maxNeverUsedIntervalMs=0 (org.apache.zookeeper.server.ContainerManager)
[2022-04-10 08:43:54,964] INFO ZooKeeper audit is disabled. (org.apache.zookeeper.audit.ZKAuditProvider)
```

It appears that it will now run indefinitely until I ctrl+c it, so I terminate the zookeeper server before splitting my terminal with putty and restarting the zookeeper server in one of the panes. I refer to [5] for tmux commands. I modify the command suggested in [3] to test if the Zookeeper server is responding and it seems fine "./zookeeper-shell.sh master:2181 ls /brokers/ids"



```
[2022-04-10 08:52:20,653] INFO Using org.apache.zookeeper.server.watch.WatchManager as watch manager (org.apache.zookeeper.server.watch.WatchManagerFactory)
[2022-04-10 08:52:20,654] INFO zookeeper.snapshotSizeFactor = 0.33 (org.apache.zookeeper.server.ZKDatabase)
[2022-04-10 08:52:20,654] INFO zookeeper.commitLogCount=500 (org.apache.zookeeper.server.ZKDatabase)
[2022-04-10 08:52:20,668] INFO zookeeper.snapshot.compression.method = CHECKED (org.apache.zookeeper.server.persistence.SnapStream)
[2022-04-10 08:52:20,669] INFO Reading snapshot /tmp/zookeeper/version-2/snapshot.0 (org.apache.zookeeper.server.persistence.FileSnap)
[2022-04-10 08:52:20,672] INFO The digest value is empty in snapshot (org.apache.zookeeper.server.DataTree)
[2022-04-10 08:52:20,676] INFO Snapshot loaded in 22 ms, highest zxid is 0x0, digest is 1371985504 (org.apache.zookeeper.server.ZKDatabase)
[2022-04-10 08:52:20,676] INFO Snapshotting: 0x0 to /tmp/zookeeper/version-2/snapshot.0 (org.apache.zookeeper.server.persistence.FileTxnSnapLog)
[2022-04-10 08:52:20,678] INFO Snapshot taken in 2 ms (org.apache.zookeeper.server.ZooKeeperServer)
[2022-04-10 08:52:20,689] INFO PrepRequestProcessor (sid:0) started, reconfigEnabled=false (org.apache.zookeeper.server.PrepRequestProcessor)
[2022-04-10 08:52:20,689] INFO zookeeper.request_throttler.shutdownTimeout = 10000 (org.apache.zookeeper.server.RequestThrottler)
[2022-04-10 08:52:20,710] INFO Using checkIntervalMs=60000 maxPerMinute=10000 maxNeverUsedIntervalMs=0 (org.apache.zookeeper.server.ContainerManager)
[2022-04-10 08:52:20,711] INFO ZooKeeper audit is disabled. (org.apache.zookeeper.audit.ZKAuditProvider)
[2022-04-10 08:53:55,725] INFO Creating new log file: log.1 (org.apache.zookeeper.server.persistence.FileTxnLog)
[2022-04-10 08:53:56,074] WARN Unexpected exception (org.apache.zookeeper.server.NIOServerCnxn)
EndOfStreamException: Unable to read additional data from client, it probably closed the socket: address = /10.140.0.6:54072, session = 0x1000028e6840000
        at org.apache.zookeeper.server.NIOServerCnxn.handleFailedRead(NIOServerCnxn.java:163)
        at org.apache.zookeeper.server.NIOServerCnxn.doIO(NIOServerCnxn.java:326)
        at org.apache.zookeeper.server.NIOServerCnxnFactory$IOWorkRequest.doWork(NIOServerCnxnFactory.java:522)
        at org.apache.zookeeper.server.WorkerService$ScheduledWorkRequest.run(WorkerService.java:154)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at java.lang.Thread.run(Thread.java:748)

EddieCFox@master:~/kafka_2.13-3.1.0/bin$ ls
connect-distributed.sh        kafka-console-consumer.sh    kafka-features.sh            kafka-reassign-partitions.sh       kafka-topics.sh                  zookeeper-server-start.sh
connect-mirror-maker.sh       kafka-console-producer.sh    kafka-get-offsets.sh         kafka-replica-verification.sh      kafka-transactions.sh            zookeeper-server-stop.sh
connect-standalone.sh         kafka-consumer-groups.sh     kafka-leader-election.sh     kafka-run-class.sh                 kafka-verifiable-consumer.sh     zookeeper-shell.sh
kafka-acls.sh                 kafka-consumer-perf-test.sh  kafka-log-dirs.sh            kafka-server-start.sh              kafka-verifiable-producer.sh
kafka-broker-api-versions.sh  kafka-delegation-tokens.sh   kafka-metadata-shell.sh      kafka-server-stop.sh               trogdor.sh
kafka-cluster.sh              kafka-delete-records.sh      kafka-mirror-maker.sh        kafka-storage.sh                   windows
kafka-configs.sh              kafka-dump-log.sh            kafka-producer-perf-test.sh  kafka-streams-application-reset.sh zookeeper-security-migration.sh
EddieCFox@master:~/kafka_2.13-3.1.0/bin$ zookeeper-shell.sh master:2181 ls /brokers/ids
zookeeper-shell.sh: command not found
EddieCFox@master:~/kafka_2.13-3.1.0/bin$ ./zookeeper-shell.sh master:2181 ls /brokers/ids
Connecting to master:2181

WATCHER::

WatchedEvent state:SyncConnected type:None path:null
Node does not exist: /brokers/ids
[2022-04-10 08:53:55,756] ERROR Exiting JVM with code 1 (org.apache.zookeeper.util.ServiceUtils)
EddieCFox@master:~/kafka_2.13-3.1.0/bin$
```

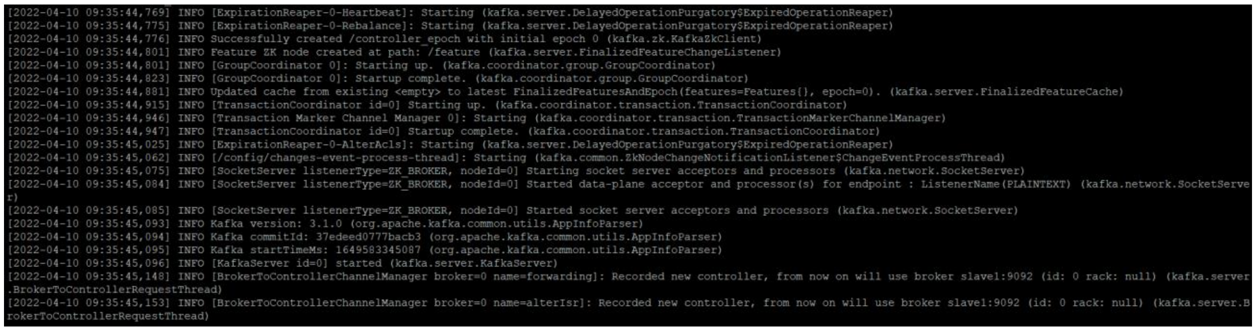
It says "Node does not exist" because we have not configured the broker servers yet.

Table of properties I am configuring on each node.

slave1	slave2
broker.id=0	broker.id=1
offsets.topic.num.partitions=2	offsets.topic.num.partitions=2
transaction.state.log.num.partitions=2	transaction.state.log.num.partitions=2
num.partitions=2	num.partitions=2
zookeeper.connect=master:2181	zookeeper.connect=master:2181
offsets.topic.replication.factor=2	offsets.topic.replication.factor=2
transaction.state.log.replication.factor=2	transaction.state.log.replication.factor=2
transaction.state.log.min.isr=2	transaction.state.log.min.isr=2
default.replication.factor=2	default.replication.factor=2
config.storage.replication.factor=2	config.storage.replication.factor=2
offset.storage.replication.factor=2	offset.storage.replication.factor=2
status.storage.replication.factor=2	status.storage.replication.factor=2
errors.deadletterqueue.topic.replication.factor =2	errors.deadletterqueue.topic.replication.factor =2

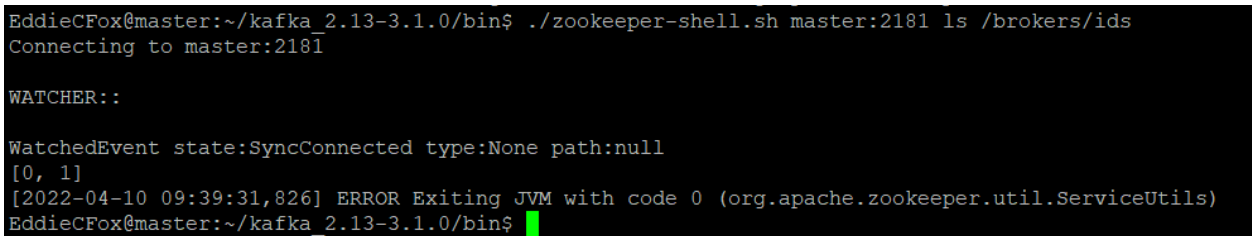
Besides the partition properties mentioned before, we also need to ensure that any replication factors are no more than 2, meaning there is no more than 2 copies of data, because we only have 2 brokers. It seems the replication factor for many things are set to 3 by default, but we are required to have 2 brokers, so I need to change all these properties. Using [4] and [3], I create the table above. Besides the partition numbers and replication factors, we have zookeeper.connect=master:2181 so the brokers are aware of the zookeeper server. The properties are basically the same for both except broker.id=0 for slave1 and broker.id=1 for slave2.

I create tmux tabs in slave1 and slave2, then run "./kafka-server-start.sh ~/kafka_2.13-3.1.0/config/server.properties" based on [2] and my previous experience with starting the Zookeeper server.



```
[2022-04-10 09:35:44,769] INFO [ExpirationReaper-0-Heartbeat]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2022-04-10 09:35:44,775] INFO [ExpirationReaper-0-Rebalance]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2022-04-10 09:35:44,776] INFO Successfully created /controller_epoch with initial epoch 0 (kafka.zk.KafkaZkClient)
[2022-04-10 09:35:44,801] INFO Feature ZK node created at path: /feature (kafka.server.FinalizedFeatureChangeListener)
[2022-04-10 09:35:44,801] INFO [GroupCoordinator 0]: Starting up. (kafka.coordinator.group.GroupCoordinator)
[2022-04-10 09:35:44,823] INFO [GroupCoordinator 0]: Startup complete. (kafka.coordinator.group.GroupCoordinator)
[2022-04-10 09:35:44,881] INFO Updated cache from existing <empty> to latest FinalizedFeaturesAndEpoch(features=Features{}, epoch=0). (kafka.server.FinalizedFeatureCache)
[2022-04-10 09:35:44,915] INFO [TransactionCoordinator id=0] Starting up. (kafka.coordinator.transaction.TransactionCoordinator)
[2022-04-10 09:35:44,946] INFO [Transaction Marker Channel Manager 0]: Starting (kafka.coordinator.transaction.TransactionMarkerChannelManager)
[2022-04-10 09:35:44,947] INFO [TransactionCoordinator id=0] Startup complete. (kafka.coordinator.transaction.TransactionCoordinator)
[2022-04-10 09:35:45,025] INFO [ExpirationReaper-0-AlterAcls]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2022-04-10 09:35:45,062] INFO [/config/changes-event-process-thread]: Starting (kafka.common.ZkNodeChangeNotificationListener$ChangeEventProcessThread)
[2022-04-10 09:35:45,075] INFO [SocketServer listenerType=ZK_BROKER, nodeId=0] Starting socket server acceptors and processors (kafka.network.SocketServer)
[2022-04-10 09:35:45,084] INFO [SocketServer listenerType=ZK_BROKER, nodeId=0] Started data-plane acceptor and processor(s) for endpoint : ListenerName(PLAINTEXT) (kafka.network.SocketServe
r)
[2022-04-10 09:35:45,085] INFO [SocketServer listenerType=ZK_BROKER, nodeId=0] Started socket server acceptors and processors (kafka.network.SocketServer)
[2022-04-10 09:35:45,093] INFO Kafka version: 3.1.0 (org.apache.kafka.common.utils.AppInfoParser)
[2022-04-10 09:35:45,094] INFO Kafka commitId: 37edeed0777bacb3 (org.apache.kafka.common.utils.AppInfoParser)
[2022-04-10 09:35:45,095] INFO Kafka startTimeMs: 1649583345087 (org.apache.kafka.common.utils.AppInfoParser)
[2022-04-10 09:35:45,096] INFO [KafkaServer id=0] started (kafka.server.KafkaServer)
[2022-04-10 09:35:45,148] INFO [BrokerToControllerChannelManager broker=0 name=forwarding]: Recorded new controller, from now on will use broker slave1:9092 (id: 0 rack: null) (kafka.server
.BrokerToControllerRequestThread)
[2022-04-10 09:35:45,153] INFO [BrokerToControllerChannelManager broker=0 name=alterIsr]: Recorded new controller, from now on will use broker slave1:9092 (id: 0 rack: null) (kafka.server.B
rokerToControllerRequestThread)
```

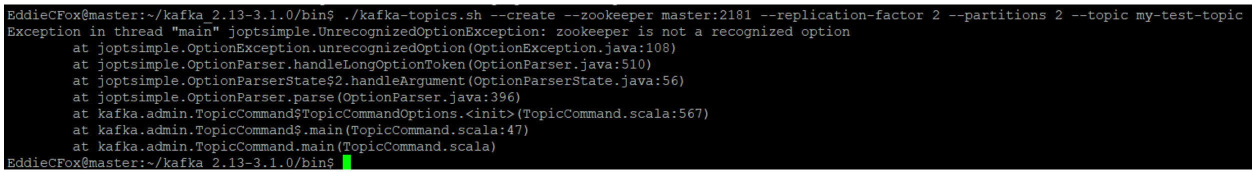
^ Above is an example of startup dialog on the slave nodes when starting a broker.



```
EddieCFox@master:~/kafka_2.13-3.1.0/bin$ ./zookeeper-shell.sh master:2181 ls /brokers/ids
Connecting to master:2181

WATCHER::

WatchedEvent state:SyncConnected type:None path:null
[0, 1]
[2022-04-10 09:39:31,826] ERROR Exiting JVM with code 0 (org.apache.zookeeper.util.ServiceUtils)
EddieCFox@master:~/kafka_2.13-3.1.0/bin$
```

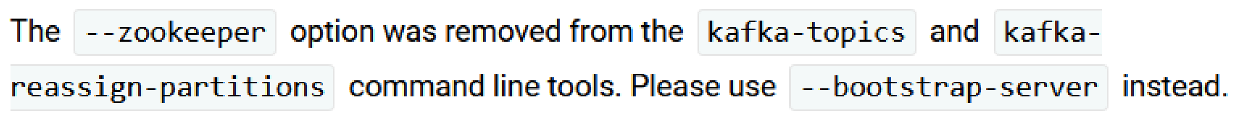
Running the ls /brokers/ids command again, we can see that instead of "Node does not exist", it says "[0,1], probably referring to the ids of the brokers.

I try running "./kafka-topics.sh --create --zookeeper master:2181 --replication-factor 2 --partitions 2 --topic my-test-topic" as suggested in the Assignment 4 PDF, but receive "zookeeper is not a recognized option" error.



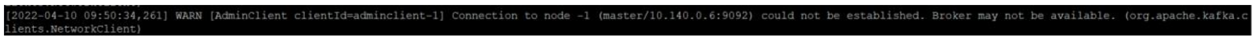
```
EddieCFox@master:~/kafka_2.13-3.1.0/bin$ ./kafka-topics.sh --create --zookeeper master:2181 --replication-factor 2 --partitions 2 --topic my-test-topic
Exception in thread "main" joptsimple.UnrecognizedOptionException: zookeeper is not a recognized option
        at joptsimple.OptionException.unrecognizedOption(OptionException.java:108)
        at joptsimple.OptionParser.handleLongOptionToken(OptionParser.java:510)
        at joptsimple.OptionParserState$2.handleArgument(OptionParserState.java:56)
        at joptsimple.OptionParser.parse(OptionParser.java:396)
        at kafka.admin.TopicCommand$TopicCommandOptions.<init>(TopicCommand.scala:567)
        at kafka.admin.TopicCommand$.main(TopicCommand.scala:47)
        at kafka.admin.TopicCommand.main(TopicCommand.scala)
EddieCFox@master:~/kafka_2.13-3.1.0/bin$
```

After looking at [4], it notes:



The `--zookeeper` option was removed from the `kafka-topics` and `kafka-reassign-partitions` command line tools. Please use `--bootstrap-server` instead.

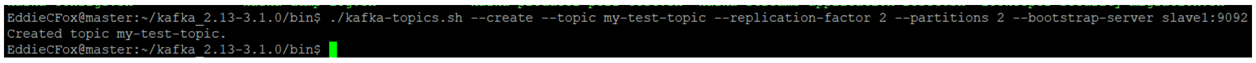
I tried running "./kafka-topics.sh --create --topic my-test-topic --replication-factor 2 --partitions 2 --bootstrap-server master:9092 "



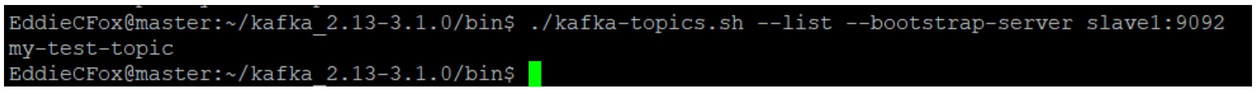
```
[2022-04-10 09:50:34,261] WARN [AdminClient clientId=adminclient-1] Connection to node -1 (master/10.140.0.6:9092) could not be established. Broker may not be available. (org.apache.kafka.c
lients.NetworkClient)
```

Based on [6], I tried modifying the bootstrap.servers property inside the server.properties file of slave1 and slave2 to say master:9092, but when restarting all the servers, I got the same error.

After doing some further reading of [6] and [7], I realized I had to put the slave nodes instead, it was looking for the brokers, not the zookeeper / master. I set the bootstrap.servers property to slave1:9092,slave2:9092 and it worked when I ran "./kafka-topics.sh --create --topic my-test-topic --replication-factor 2 --partitions 2 --bootstrap-server slave1:9092"

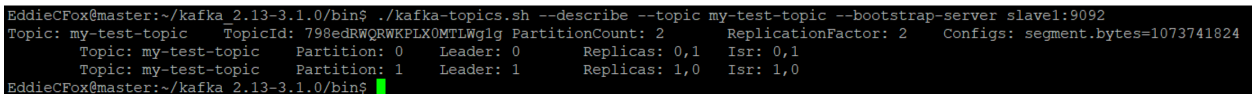


```
EddieCFox@master:~/kafka_2.13-3.1.0/bin$ ./kafka-topics.sh --create --topic my-test-topic --replication-factor 2 --partitions 2 --bootstrap-server slave1:9092
Created topic my-test-topic.
EddieCFox@master:~/kafka_2.13-3.1.0/bin$
```



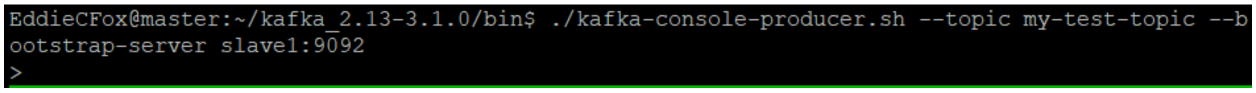
```
EddieCFox@master:~/kafka_2.13-3.1.0/bin$ ./kafka-topics.sh --list --bootstrap-server slave1:9092
my-test-topic
EddieCFox@master:~/kafka_2.13-3.1.0/bin$
```

I am able to confirm the topic when listing it with "./kafka-topics.sh --list --bootstrap-server slave1:9092"



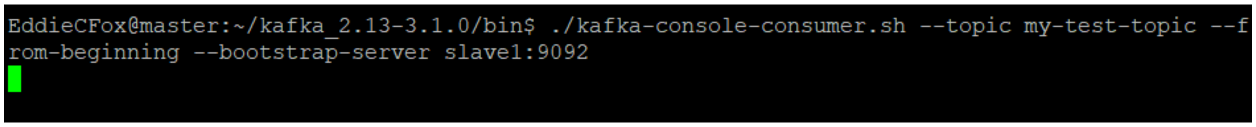
```
EddieCFox@master:~/kafka_2.13-3.1.0/bin$ ./kafka-topics.sh --describe --topic my-test-topic --bootstrap-server slave1:9092
Topic: my-test-topic    TopicId: 798edRWQRWKPLX0MTLWg1g PartitionCount: 2       ReplicationFactor: 2    Configs: segment.bytes=1073741824
        Topic: my-test-topic    Partition: 0    Leader: 0       Replicas: 0,1   Isr: 0,1
        Topic: my-test-topic    Partition: 1    Leader: 1       Replicas: 1,0   Isr: 1,0
EddieCFox@master:~/kafka_2.13-3.1.0/bin$
```

Using the modified command from [2], "./kafka-topics.sh --describe --topic my-test-topic --bootstrap-server slave1:9092" I am able to confirm there is 2 partitions for the topic and 2 replicas, one on each of the brokers.



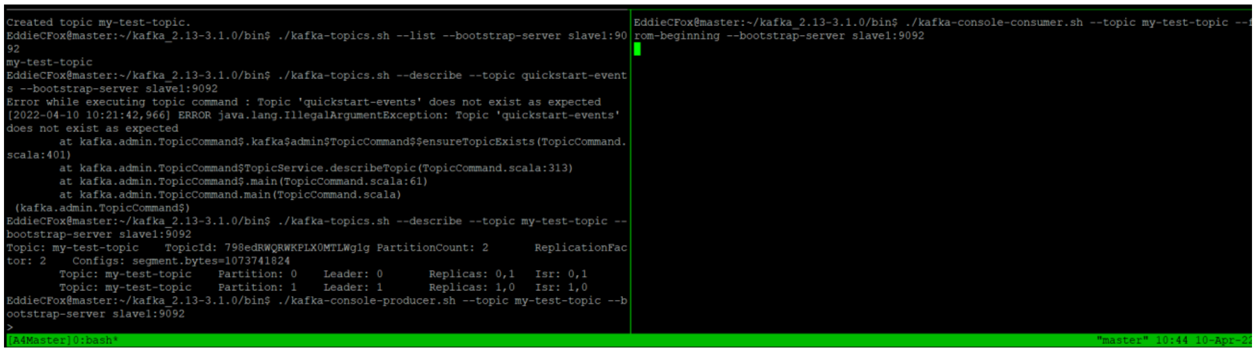
```
EddieCFox@master:~/kafka_2.13-3.1.0/bin$ ./kafka-console-producer.sh --topic my-test-topic --b
ootstrap-server slave1:9092
>
```

Based on [2], I run "./kafka-console-producer.sh --topic my-test-topic --bootstrap-server slave1:9092" to start the producer for the test topic. I am now able to type what I want.



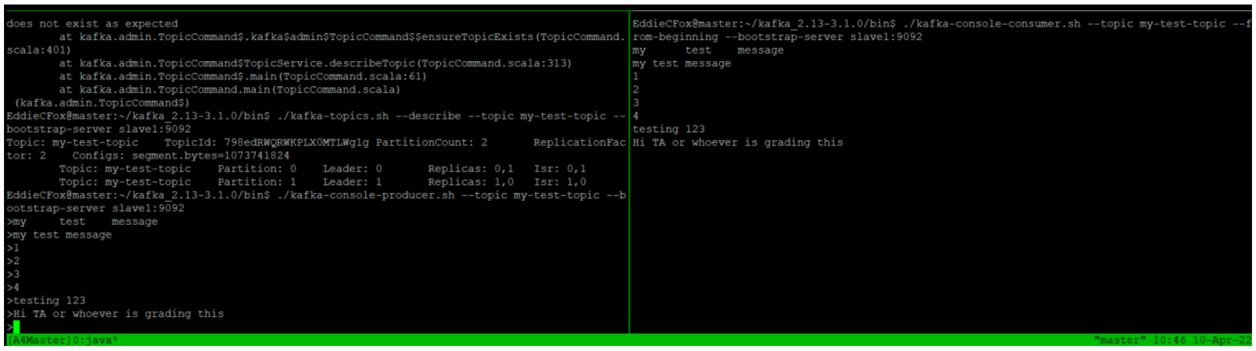
```
EddieCFox@master:~/kafka_2.13-3.1.0/bin$ ./kafka-console-consumer.sh --topic my-test-topic --f
rom-beginning --bootstrap-server slave1:9092
```

I use "./kafka-console-consumer.sh --topic my-test-topic --from-beginning --bootstrap-server slave1:9092" to start the consumer for the test topic. I have it in split screen so that I can type messages into the producer on the left and immediately see output read into the consumer on the right.



```
Created topic my-test-topic.
EddieCFox@master:~/kafka_2.13-3.1.0/bin$ ./kafka-topics.sh --list --bootstrap-server slave1:90
92
my-test-topic
EddieCFox@master:~/kafka_2.13-3.1.0/bin$ ./kafka-topics.sh --describe --topic quickstart-event
s --bootstrap-server slave1:9092
Error while executing topic command : Topic 'quickstart-events' does not exist as expected
[2022-04-10 10:21:42,966] ERROR java.lang.IllegalArgumentException: Topic 'quickstart-events'
does not exist as expected
        at kafka.admin.TopicCommand$.kafka$admin$TopicCommand$$ensureTopicExists(TopicCommand.
scala:401)
        at kafka.admin.TopicCommand$TopicService.describeTopic(TopicCommand.scala:313)
        at kafka.admin.TopicCommand$.main(TopicCommand.scala:61)
        at kafka.admin.TopicCommand.main(TopicCommand.scala)
 (kafka.admin.TopicCommand$)
EddieCFox@master:~/kafka_2.13-3.1.0/bin$ ./kafka-topics.sh --describe --topic my-test-topic --
bootstrap-server slave1:9092
Topic: my-test-topic    TopicId: 798edRWQRWKPLX0MTLWg1g PartitionCount: 2       ReplicationFac
tor: 2    Configs: segment.bytes=1073741824
        Topic: my-test-topic    Partition: 0    Leader: 0       Replicas: 0,1   Isr: 0,1
        Topic: my-test-topic    Partition: 1    Leader: 1       Replicas: 1,0   Isr: 1,0
EddieCFox@master:~/kafka_2.13-3.1.0/bin$ ./kafka-console-producer.sh --topic my-test-topic --b
ootstrap-server slave1:9092
>
[A4Master]0:bash*
```

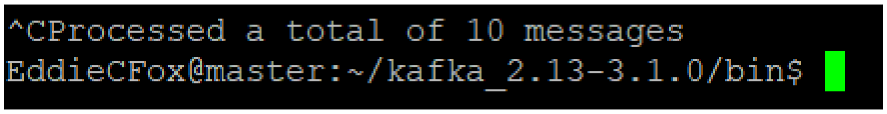
```
EddieCFox@master:~/kafka_2.13-3.1.0/bin$ ./kafka-console-consumer.sh --topic my-test-topic --f
rom-beginning --bootstrap-server slave1:9092
"master" 10:44 10-Apr-22
```



```
does not exist as expected
        at kafka.admin.TopicCommand$.kafka$admin$TopicCommand$$ensureTopicExists(TopicCommand.
scala:401)
        at kafka.admin.TopicCommand$TopicService.describeTopic(TopicCommand.scala:313)
        at kafka.admin.TopicCommand$.main(TopicCommand.scala:61)
        at kafka.admin.TopicCommand.main(TopicCommand.scala)
 (kafka.admin.TopicCommand$)
EddieCFox@master:~/kafka_2.13-3.1.0/bin$ ./kafka-topics.sh --describe --topic my-test-topic --
bootstrap-server slave1:9092
Topic: my-test-topic    TopicId: 798edRWQRWKPLX0MTLWg1g PartitionCount: 2       ReplicationFac
tor: 2    Configs: segment.bytes=1073741824
        Topic: my-test-topic    Partition: 0    Leader: 0       Replicas: 0,1   Isr: 0,1
        Topic: my-test-topic    Partition: 1    Leader: 1       Replicas: 1,0   Isr: 1,0
EddieCFox@master:~/kafka_2.13-3.1.0/bin$ ./kafka-console-producer.sh --topic my-test-topic --b
ootstrap-server slave1:9092
>my     test    message
>my test message
>1
>2
>3
>4
>testing 123
>Hi TA or whoever is grading this
>
[A4Master]0:java*
```

```
EddieCFox@master:~/kafka_2.13-3.1.0/bin$ ./kafka-console-consumer.sh --topic my-test-topic --f
rom-beginning --bootstrap-server slave1:9092
my      test    message
my test message
1
2
3
4
testing 123
Hi TA or whoever is grading this
"master" 10:46 10-Apr-22
```

On the left, I type "my test message" and "my   test   message" into the producer and receive identical output on the consumer within about 2-3 seconds.



```
^CProcessed a total of 10 messages
EddieCFox@master:~/kafka_2.13-3.1.0/bin$
```

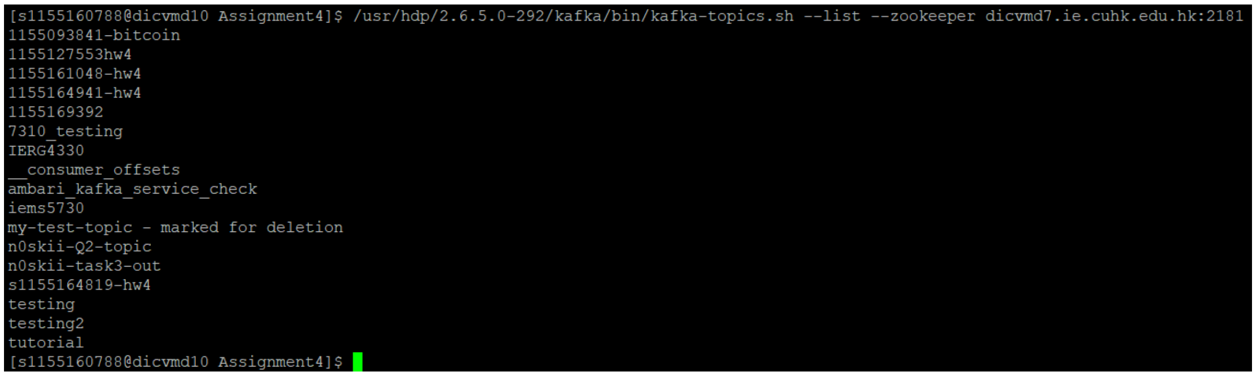
Upon pressing ctrl+c on the consumer, it tells me how many messages were processed total.

Having satisfied the requirements of this question, I consider it complete.

Assignment continued on next page.

Question 2 Section A: Count the Most Frequent Hashtags with Spark RDD Streaming

Final code on Page 15, final submission on page 17



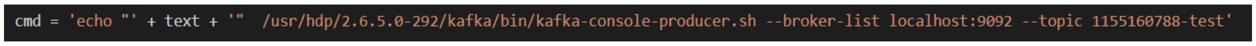
```
[s1155160788@dicvmd10 Assignment4]$ /usr/hdp/2.6.5.0-292/kafka/bin/kafka-topics.sh --list --zookeeper dicvmd7.ie.cuhk.edu.hk:2181
1155093841-bitcoin
1155127553hw4
1155161048-hw4
1155164941-hw4
1155169392
7310_testing
IERG4330
__consumer_offsets
ambari_kafka_service_check
iems5730
my-test-topic - marked for deletion
n0skii-Q2-topic
n0skii-task3-out
s1155164819-hw4
testing
testing2
tutorial
[s1155160788@dicvmd10 Assignment4]$
```

I first confirm that commands from the IE DIC Kafka cluster tutorial[8] work. It appears the IE DIC cluster is using an earlier version of Kafka that still uses commands with zookeeper instead of bootstrap-server specified. I am able to successfully list topics here using "/usr/hdp/2.6.5.0-292/kafka/bin/kafka-topics.sh --list --zookeeper dicvmd7.ie.cuhk.edu.hk:2181"

The first step will be to get the producer working to ingest the dataset appropriately. I download the dataset from [9] and upload it to the IE DIC cluster through Filezilla.

After examining the provided kafka_producer.py [10], I change the name of the dataset to "bitcoin_twitter.txt" and reupload it to the server.

I upload kafka_producer.py to the server mostly unmodified and test topic creation and running the producer.py and regular consumer to gain familiarity with the command and confirm the current implementation.



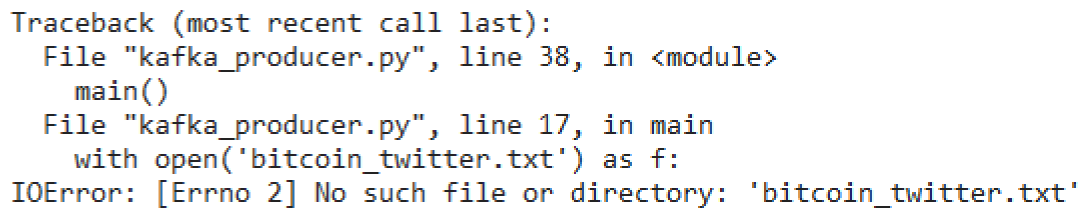
```
cmd = 'echo "' + text + '"  /usr/hdp/2.6.5.0-292/kafka/bin/kafka-console-producer.sh --broker-list localhost:9092 --topic 1155160788-test'
```

I only change line 27 for now so that I can run the script from anywhere rather than using a relative directory path that forces me to be in the kafka folder, and I change the topic name to match with I plan to use.

I create a test topic called 1155160788-test with "/usr/hdp/2.6.5.0-292/kafka/bin/kafka-topics.sh --create --zookeeper dicvmd7.ie.cuhk.edu.hk:2181 --replication-factor 2 -partitions 2 --topic 1155160788-test" [8].

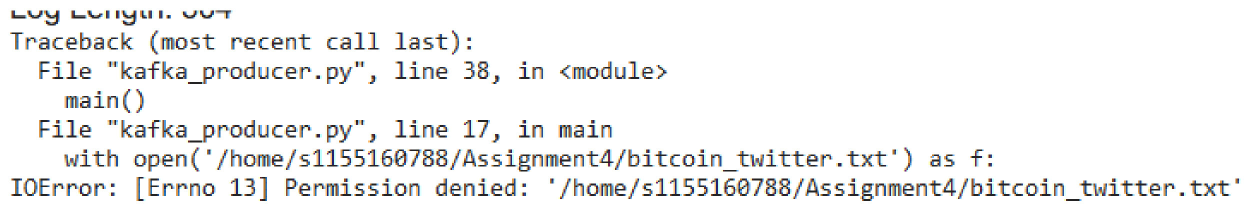
I separate my tabs with tmux to run the consumer in one pane, then run the producer.py in the other one using "spark-submit --master yarn --deploy-mode cluster --packages org.apache.spark:spark-streaming-kafka-0-8_2.11:2.3.0 kafka_producer.py" from [8].

When I first try this, I receive errors.



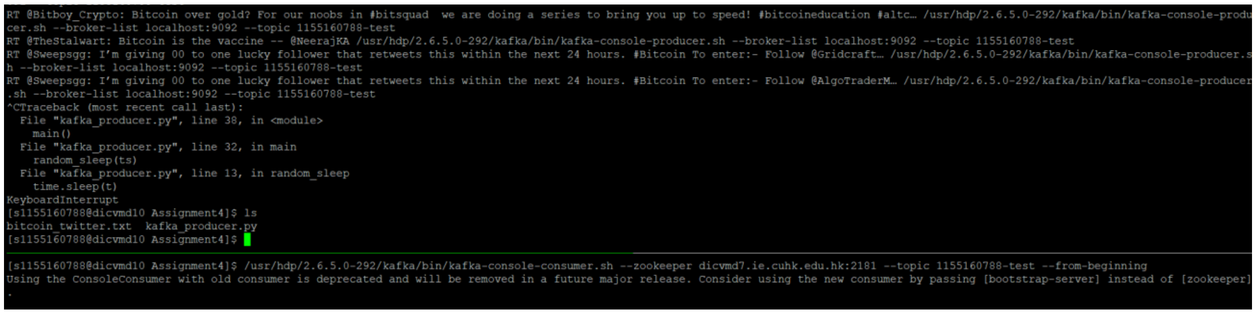
```
Traceback (most recent call last):
  File "kafka_producer.py", line 38, in <module>
    main()
  File "kafka_producer.py", line 17, in main
    with open('bitcoin_twitter.txt') as f:
IOError: [Errno 2] No such file or directory: 'bitcoin_twitter.txt'
```

The above is from the stdout log of the failed application attempt, where it says there is no such file as bitcoin_twitter.txt. I ran it in my Assignment4 directory where the only files were the producer.py and the bitcoin_twitter.txt. I decide to upload the bitcoin_twitter.txt to the HDFS and try again. I still receive the same error.



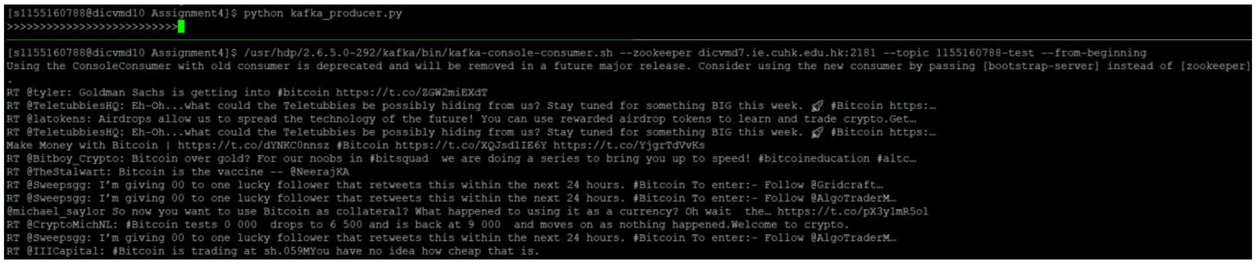
```
Traceback (most recent call last):
  File "kafka_producer.py", line 38, in <module>
    main()
  File "kafka_producer.py", line 17, in main
    with open('/home/s1155160788/Assignment4/bitcoin_twitter.txt') as f:
IOError: [Errno 13] Permission denied: '/home/s1155160788/Assignment4/bitcoin_twitter.txt'
```

After changing it to the absolute local path [11], I get another error saying “Permission denied”. I use “chmod 666 bitcoin_twitter.txt” in the Assignment 4 directory but still receive the same error.



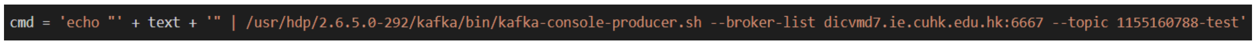
```
RT @Bitboy_Crypto: Bitcoin over gold? For our noobs in #bitsquad  we are doing a series to bring you up to speed! #bitcoineducation #altc… /usr/hdp/2.6.5.0-292/kafka/bin/kafka-console-produ
cer.sh --broker-list localhost:9092 --topic 1155160788-test
RT @TheStalwart: Bitcoin is the vaccine -- @NeerajKA /usr/hdp/2.6.5.0-292/kafka/bin/kafka-console-producer.sh --broker-list localhost:9092 --topic 1155160788-test
RT @Sweepsgg: I’m giving 00 to one lucky follower that retweets this within the next 24 hours. #Bitcoin To enter:- Follow @Gridcraft… /usr/hdp/2.6.5.0-292/kafka/bin/kafka-console-producer.s
h --broker-list localhost:9092 --topic 1155160788-test
RT @Sweepsgg: I’m giving 00 to one lucky follower that retweets this within the next 24 hours. #Bitcoin To enter:- Follow @AlgoTraderM… /usr/hdp/2.6.5.0-292/kafka/bin/kafka-console-producer
.sh --broker-list localhost:9092 --topic 1155160788-test
^CTraceback (most recent call last):
  File "kafka_producer.py", line 38, in <module>
    main()
  File "kafka_producer.py", line 32, in main
    random_sleep(ts)
  File "kafka_producer.py", line 13, in random_sleep
    time.sleep(t)
KeyboardInterrupt
[s1155160788@dicvmd10 Assignment4]$ ls
bitcoin_twitter.txt  kafka_producer.py
[s1155160788@dicvmd10 Assignment4]$ 

[s1155160788@dicvmd10 Assignment4]$ /usr/hdp/2.6.5.0-292/kafka/bin/kafka-console-consumer.sh --zookeeper dicvmd7.ie.cuhk.edu.hk:2181 --topic 1155160788-test --from-beginning
Using the ConsoleConsumer with old consumer is deprecated and will be removed in a future major release. Consider using the new consumer by passing [bootstrap-server] instead of [zookeeper]
.
```

When trying to run the producer script with python instead of spark submit, it seemed to work and print out the tweets, however it would not be sent to the consumer. I also saw that it was echoing the twitter text followed by the broker list command instead of actually using the command. I looked at the tutorial [8] and used the line under the producer.py there that included the pipeline in the echo statement.



```
[s1155160788@dicvmd10 Assignment4]$ python kafka_producer.py
>>>>>>>>>>>>>>>>>>>>>>>>>>>

[s1155160788@dicvmd10 Assignment4]$ /usr/hdp/2.6.5.0-292/kafka/bin/kafka-console-consumer.sh --zookeeper dicvmd7.ie.cuhk.edu.hk:2181 --topic 1155160788-test --from-beginning
Using the ConsoleConsumer with old consumer is deprecated and will be removed in a future major release. Consider using the new consumer by passing [bootstrap-server] instead of [zookeeper]
.
RT @tyler: Goldman Sachs is getting into #bitcoin https://t.co/ZGW2miEXdT
RT @TeletubbiesHQ: Eh-Oh...what could the Teletubbies be possibly hiding from us? Stay tuned for something BIG this week. 🚀 #Bitcoin https:…
RT @latokens: Airdrops allow us to spread the technology of the future! You can use rewarded airdrop tokens to learn and trade crypto.Get…
RT @TeletubbiesHQ: Eh-Oh...what could the Teletubbies be possibly hiding from us? Stay tuned for something BIG this week. 🚀 #Bitcoin https:…
Make Money with Bitcoin | https://t.co/dYNKC0nnsz #Bitcoin https://t.co/XQJsdlIE6Y https://t.co/YjgrTdVvKs
RT @Bitboy_Crypto: Bitcoin over gold? For our noobs in #bitsquad  we are doing a series to bring you up to speed! #bitcoineducation #altc…
RT @TheStalwart: Bitcoin is the vaccine -- @NeerajKA
RT @Sweepsgg: I’m giving 00 to one lucky follower that retweets this within the next 24 hours. #Bitcoin To enter:- Follow @Gridcraft…
RT @Sweepsgg: I’m giving 00 to one lucky follower that retweets this within the next 24 hours. #Bitcoin To enter:- Follow @AlgoTraderM…
@michael_saylor So now you want to use Bitcoin as collateral? What happened to using it as a currency? Oh wait  the… https://t.co/pX3ylmR5ol
RT @CryptoMichNL: #Bitcoin tests 0 000  drops to 6 500 and is back at 9 000  and moves on as nothing happened.Welcome to crypto.
RT @Sweepsgg: I’m giving 00 to one lucky follower that retweets this within the next 24 hours. #Bitcoin To enter:- Follow @AlgoTraderM…
RT @IIICapital: #Bitcoin is trading at sh.059MYou have no idea how cheap that is.
```

After uploading the new producer.py, we finally have the desired behavior. The top pane is the producer script and the bottom pane is the output from the consumer.



```
cmd = 'echo "' + text + '" | /usr/hdp/2.6.5.0-292/kafka/bin/kafka-console-producer.sh --broker-list dicvmd7.ie.cuhk.edu.hk:6667 --topic 1155160788-test'
```

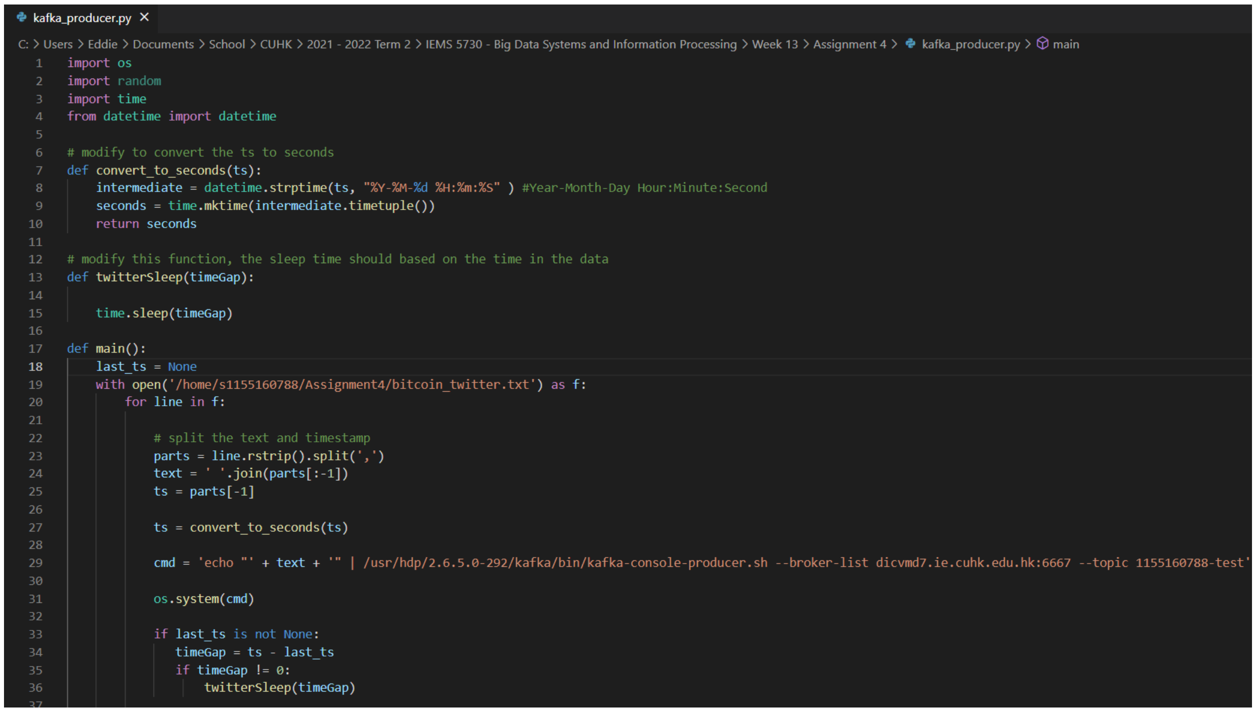
Here is the final line 27 to get the producer.py work with vanilla functionality. I realized that the spark-submit of a python file was for the consumer, not for the producer script, which is probably why there was issues getting it to work that way. When I rewatched the zoom recording, I realized they were using python to run the producer and spark-submit for the consumer.

I now need to program the code for the actual producer, and I will evaluate it using the standard consumer to make sure the messages are coming in at the right rate.

I refer to [12] to understand the slicing syntax used in the original code.

I know that the timestamp is in the format 2021-03-31 19:46:11 and need to modify the convert_to_seconds() function. I use [13] to figure out how to convert a timestamp string into the number of seconds, then look at [14] to confirm I have the right formatting.

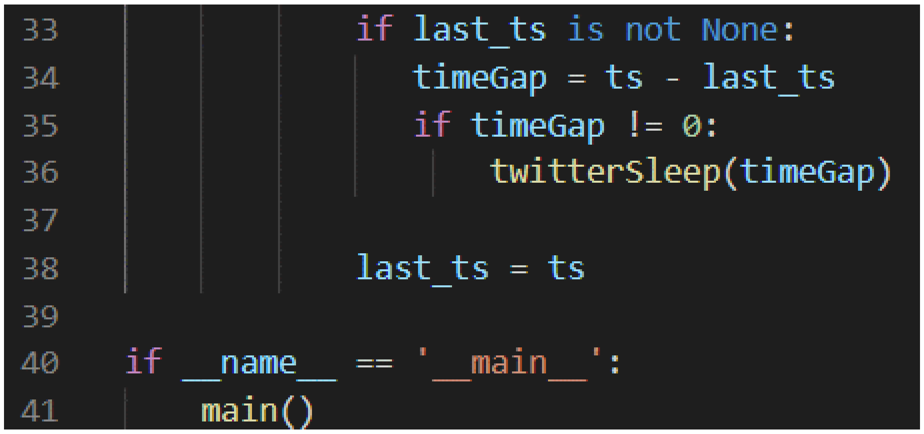
The final producer code is as follows:



kafka_producer.py

C: > Users > Eddie > Documents > School > CUHK > 2021 - 2022 Term 2 > IEMS 5730 - Big Data Systems and Information Processing > Week 13 > Assignment 4 > kafka_producer.py > main

```python
1   import os
2   import random
3   import time
4   from datetime import datetime
5
6   # modify to convert the ts to seconds
7   def convert_to_seconds(ts):
8       intermediate = datetime.strptime(ts, "%Y-%M-%d %H:%m:%S" ) #Year-Month-Day Hour:Minute:Second
9       seconds = time.mktime(intermediate.timetuple())
10      return seconds
11
12  # modify this function, the sleep time should based on the time in the data
13  def twitterSleep(timeGap):
14
15      time.sleep(timeGap)
16
17  def main():
18      last_ts = None
19      with open('/home/s1155160788/Assignment4/bitcoin_twitter.txt') as f:
20          for line in f:
21
22              # split the text and timestamp
23              parts = line.rstrip().split(',')
24              text = ' '.join(parts[:-1])
25              ts = parts[-1]
26
27              ts = convert_to_seconds(ts)
28
29              cmd = 'echo "' + text + '" | /usr/hdp/2.6.5.0-292/kafka/bin/kafka-console-producer.sh --broker-list dicvmd7.ie.cuhk.edu.hk:6667 --topic 1155160788-test'
30
31              os.system(cmd)
32
33              if last_ts is not None:
34                  timeGap = ts - last_ts
35                  if timeGap != 0:
36                      twitterSleep(timeGap)
```



```python
33              if last_ts is not None:
34                  timeGap = ts - last_ts
35                  if timeGap != 0:
36                      twitterSleep(timeGap)
37
38              last_ts = ts
39
40  if __name__ == '__main__':
41      main()
```

I need to import datetime for the convert_to_seconds function to work. On the first run, ts is none.

The code reads in the tweet and breaks it into parts based on commas, setting all but the last as the timestamp, and setting the last one as the timestamp. The timestamp is then converted to seconds, and the text is sent to the kafka topic through the cmd line 29 and 31.

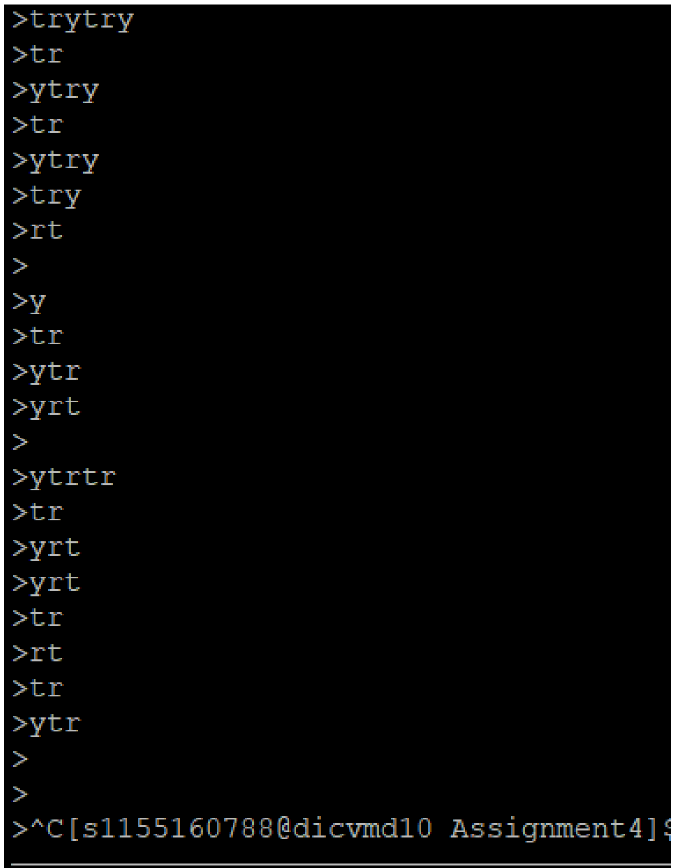
On the first run, the code moves onto the next Tweet because last_ts is none, but not before setting the last_ts variable to equal ts, which has already been converted into seconds by this point.

On future runs, the code runs the same, but at the end, it takes the last timestamp (in seconds) and subtracts it from the current timestamp before sleeping for as long as the difference is (assuming the difference in timestamp is not 0). If the second tweet is two seconds in the future and the first timestamp is 123456, then last_ts will be set to 123456 after the first tweet and ts will become 123458, leading to a sleep of (123458 -123456) = 2 seconds, exactly as we would expect. After the sleep, the timestamp becomes the last timestamp and the cycle continues.

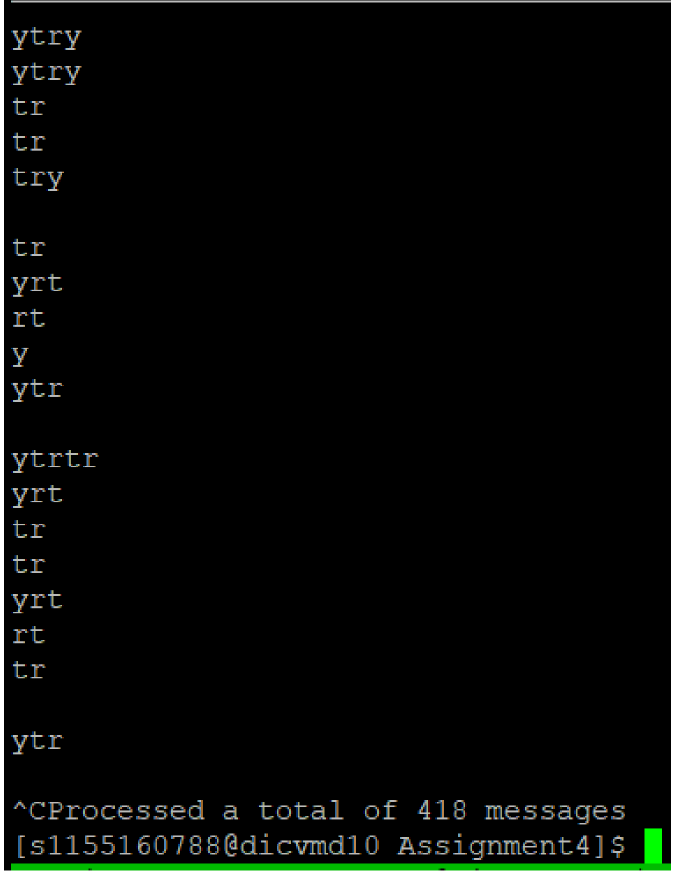
This is a little hard to verify. What I ended up doing is looking at the raw Twitter txt file and counting the number of Tweets from 19:46:11 to 19:47:11 using the line numbers (one minute, total number of Tweets is about 222). I then start the consumer, then the producer and set a time of 1 minute before terminating the producer, followed by terminating the consumer. The consumer, when terminated, shows how many messages it processed, and this should be close to 222.

At first it only processed 46 messages in 1 minute, a little less than 1 per second so I felt it was not working correctly. I tried commenting out various sections of the code but the time was still around the same, even though it should be processing much faster without the sleep commands.

Screenshot on next page.

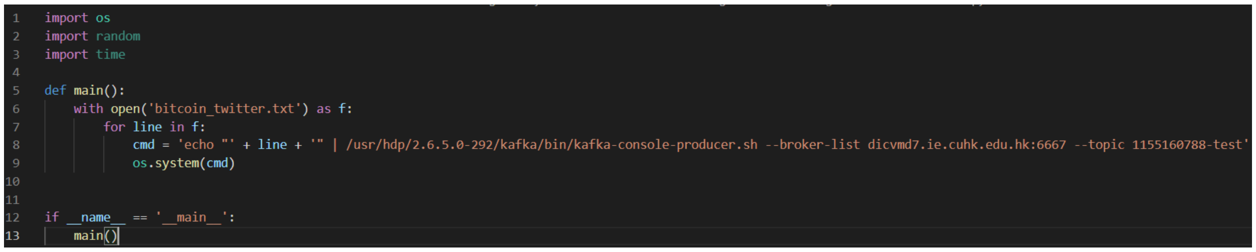


```
>trytry
>tr
>ytry
>tr
>ytry
>try
>rt
>
>y
>tr
>ytr
>yrt
>
>ytrtr
>tr
>yrt
>yrt
>tr
>rt
>tr
>ytr
>
>
>^C[s1155160788@dicvmd10 Assignment4]
```



```
ytry
ytry
tr
tr
try

tr
yrt
rt
y
ytr

ytrtr
yrt
tr
tr
yrt
rt
tr

ytr

^CProcessed a total of 418 messages
[s1155160788@dicvmd10 Assignment4]$
```

I tried typing as fast as I could and pressing enter using a manual producer ("/usr/hdp/2.6.5.0-292/kafka/bin/kafka-console-producer.sh --broker-list dicvmd7.ie.cuhk.edu.hk:6667 --topic 1155160788-test") and it was able to process about 7 messages per second, meaning the issue is not on the consumer but the producer.



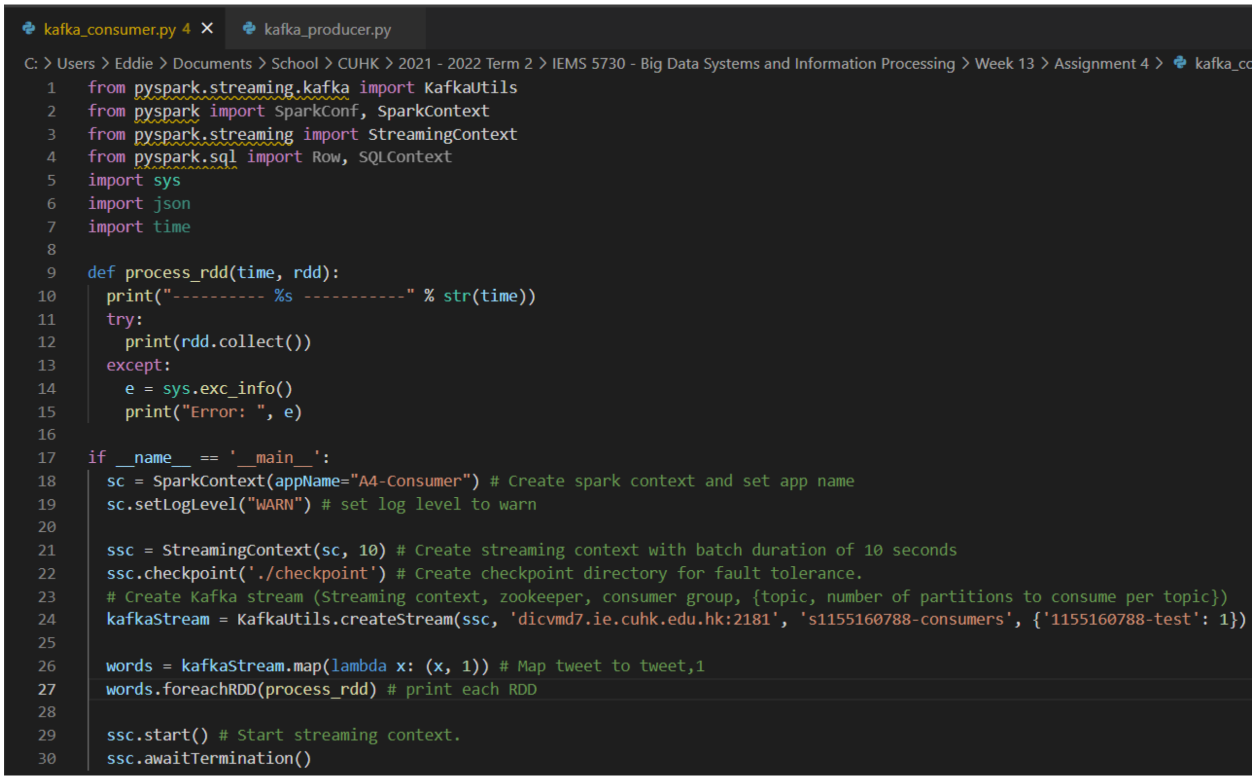
```
import os
import random
import time

def main():
    with open('bitcoin_twitter.txt') as f:
        for line in f:
            cmd = 'echo "' + line + '" | /usr/hdp/2.6.5.0-292/kafka/bin/kafka-console-producer.sh --broker-list dicvmd7.ie.cuhk.edu.hk:6667 --topic 1155160788-test'
            os.system(cmd)


if __name__ == '__main__':
    main()
```

I tried a stripped down implementation with only reading the line in and the cmd to echo it to the kafka producer, but this only achieved a rate of 105 tweets per minute, not enough to satisfy the 222 per minute needed for the content to be read in real time. When adding in the timestamp stripping, processing, and the sleep commands whenever the timestamp of the next tweet is different, the producer only does around 45 tweets per minute.

The TA's later clarified that the producer is limited in its rate because it takes time for the producer to communicate with the brokers. Receiving about 45 tweets per minute is further validation that I implemented the sleep functionality correctly. The base rate without the sleep commands is 60 tweets per minute. There are roughly 4 tweets per second. Whenever the timestamp changes, the producer will sleep for 1 second. Since this happens about every 4th tweet, we would expect there to be a 1 second sleep 15 times per minute. Without sleep commands, it can produce about 1 tweet per second, but 15 seconds each minute are spent on sleeping, so we end up with approximately 45 expected tweets with proper sleeping, which is true.

I start with the kafka_consumer.py base provided by the IE DIC tutorial [8]. I do research to understand the various lines used, such as referring to [15] to learn more about checkpointing and confirming the checkpointing syntax. The tutorial seems to link to a Kafka 0.10 integration guide, but the kafka_consumer.py in the tutorial, as well as the tutorial they link to use Kafka 0.8[16], so I refer to [17] going forward.

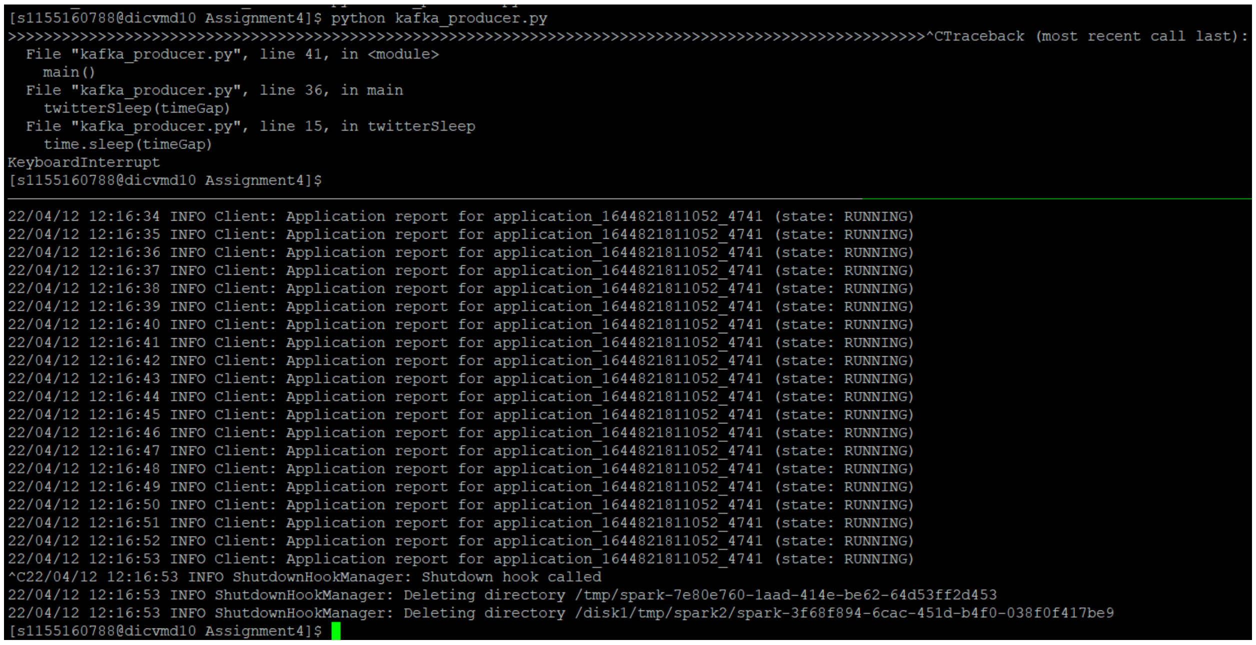
Screenshot on next page (of the initial code tested, not final code)


kafka_consumer.py 4 ✕ | kafka_producer.py

C: > Users > Eddie > Documents > School > CUHK > 2021 - 2022 Term 2 > IEMS 5730 - Big Data Systems and Information Processing > Week 13 > Assignment 4 > kafka_c

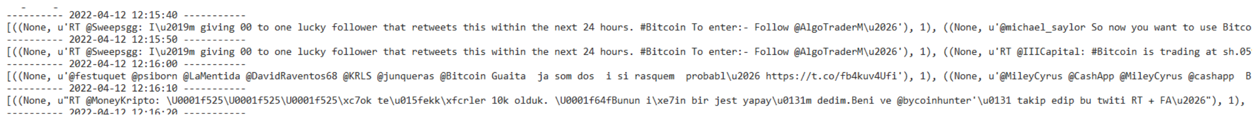
```python
from pyspark.streaming.kafka import KafkaUtils
from pyspark import SparkConf, SparkContext
from pyspark.streaming import StreamingContext
from pyspark.sql import Row, SQLContext
import sys
import json
import time

def process_rdd(time, rdd):
  print("---------- %s -----------" % str(time))
  try:
    print(rdd.collect())
  except:
    e = sys.exc_info()
    print("Error: ", e)

if __name__ == '__main__':
  sc = SparkContext(appName="A4-Consumer") # Create spark context and set app name
  sc.setLogLevel("WARN") # set log level to warn

  ssc = StreamingContext(sc, 10) # Create streaming context with batch duration of 10 seconds
  ssc.checkpoint('./checkpoint') # Create checkpoint directory for fault tolerance.
  # Create Kafka stream (Streaming context, zookeeper, consumer group, {topic, number of partitions to consume per topic})
  kafkaStream = KafkaUtils.createStream(ssc, 'dicvmd7.ie.cuhk.edu.hk:2181', 's1155160788-consumers', {'1155160788-test': 1})

  words = kafkaStream.map(lambda x: (x, 1)) # Map tweet to tweet,1
  words.foreachRDD(process_rdd) # print each RDD

  ssc.start() # Start streaming context.
  ssc.awaitTermination()
```

I first test the base consumer code from the tutorial after modifying some of the parameters to meet my situation, but not changing the RDD processing yet just to make sure it will work on the cluster without any errors.


```
[s1155160788@dicvmd10 Assignment4]$ python kafka_producer.py
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>^CTraceback (most recent call last):
  File "kafka_producer.py", line 41, in <module>
    main()
  File "kafka_producer.py", line 36, in main
    twitterSleep(timeGap)
  File "kafka_producer.py", line 15, in twitterSleep
    time.sleep(timeGap)
KeyboardInterrupt
[s1155160788@dicvmd10 Assignment4]$
```

```
22/04/12 12:16:34 INFO Client: Application report for application_1644821811052_4741 (state: RUNNING)
22/04/12 12:16:35 INFO Client: Application report for application_1644821811052_4741 (state: RUNNING)
22/04/12 12:16:36 INFO Client: Application report for application_1644821811052_4741 (state: RUNNING)
22/04/12 12:16:37 INFO Client: Application report for application_1644821811052_4741 (state: RUNNING)
22/04/12 12:16:38 INFO Client: Application report for application_1644821811052_4741 (state: RUNNING)
22/04/12 12:16:39 INFO Client: Application report for application_1644821811052_4741 (state: RUNNING)
22/04/12 12:16:40 INFO Client: Application report for application_1644821811052_4741 (state: RUNNING)
22/04/12 12:16:41 INFO Client: Application report for application_1644821811052_4741 (state: RUNNING)
22/04/12 12:16:42 INFO Client: Application report for application_1644821811052_4741 (state: RUNNING)
22/04/12 12:16:43 INFO Client: Application report for application_1644821811052_4741 (state: RUNNING)
22/04/12 12:16:44 INFO Client: Application report for application_1644821811052_4741 (state: RUNNING)
22/04/12 12:16:45 INFO Client: Application report for application_1644821811052_4741 (state: RUNNING)
22/04/12 12:16:46 INFO Client: Application report for application_1644821811052_4741 (state: RUNNING)
22/04/12 12:16:47 INFO Client: Application report for application_1644821811052_4741 (state: RUNNING)
22/04/12 12:16:48 INFO Client: Application report for application_1644821811052_4741 (state: RUNNING)
22/04/12 12:16:49 INFO Client: Application report for application_1644821811052_4741 (state: RUNNING)
22/04/12 12:16:50 INFO Client: Application report for application_1644821811052_4741 (state: RUNNING)
22/04/12 12:16:51 INFO Client: Application report for application_1644821811052_4741 (state: RUNNING)
22/04/12 12:16:52 INFO Client: Application report for application_1644821811052_4741 (state: RUNNING)
22/04/12 12:16:53 INFO Client: Application report for application_1644821811052_4741 (state: RUNNING)
^C22/04/12 12:16:53 INFO ShutdownHookManager: Shutdown hook called
22/04/12 12:16:53 INFO ShutdownHookManager: Deleting directory /tmp/spark-7e80e760-1aad-414e-be62-64d53ff2d453
22/04/12 12:16:53 INFO ShutdownHookManager: Deleting directory /disk1/tmp/spark2/spark-3f68f894-6cac-451d-b4f0-038f0f417be9
[s1155160788@dicvmd10 Assignment4]$
```

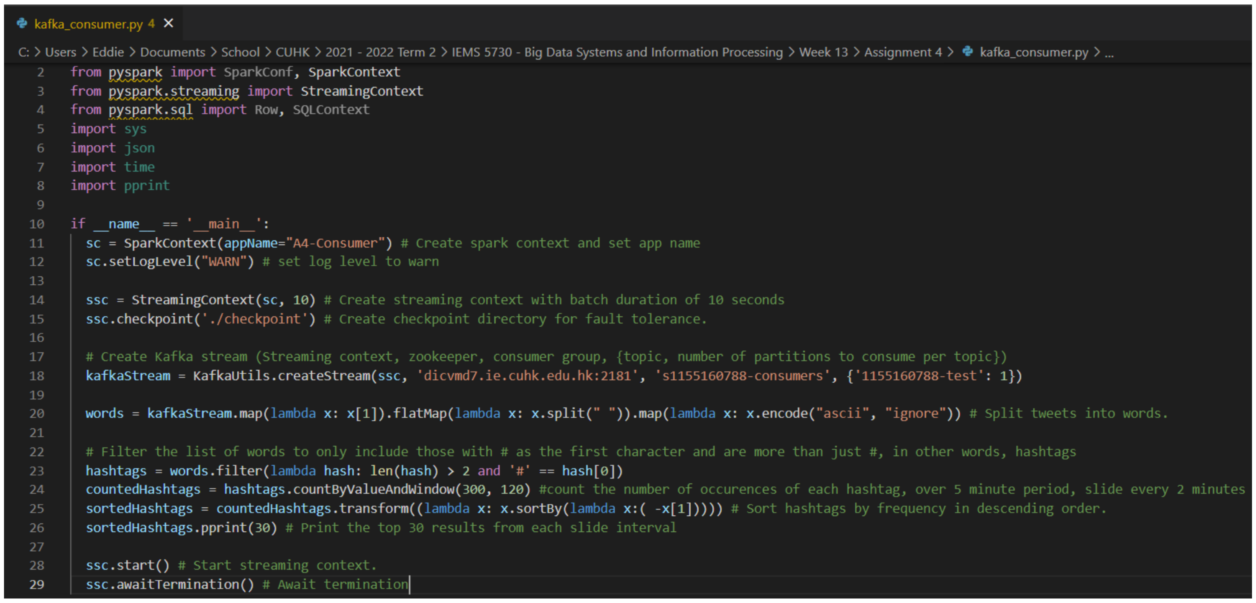
As usual, top is the producer and bottom is the consumer.



```
---------- 2022-04-12 12:15:40 -----------
[((None, u'RT @Sweepsgg: I\u2019m giving 00 to one lucky follower that retweets this within the next 24 hours. #Bitcoin To enter:- Follow @AlgoTraderM\u2026'), 1), ((None, u'@michael_saylor So now you want to use Bitco
---------- 2022-04-12 12:15:50 -----------
[((None, u'RT @Sweepsgg: I\u2019m giving 00 to one lucky follower that retweets this within the next 24 hours. #Bitcoin To enter:- Follow @AlgoTraderM\u2026'), 1), ((None, u'RT @IIICapital: #Bitcoin is trading at sh.05
---------- 2022-04-12 12:16:00 -----------
[((None, u'@festuquet @psiborn @LaMentida @DavidRaventos68 @KRLS @junqueras @Bitcoin Guaita  ja som dos  i si rasquem  probabl\u2026 https://t.co/fb4kuv4Ufi'), 1), ((None, u'@MileyCyrus @CashApp @MileyCyrus @cashapp  B
---------- 2022-04-12 12:16:10 -----------
[((None, u"RT @MoneyKripto: \U0001f525\U0001f525\U0001f525\xc7ok te\u015fekk\xfcrler 10k olduk. \U0001f64fBunun i\xe7in bir jest yapay\u0131m dedim.Beni ve @bycoinhunter'\u0131 takip edip bu twiti RT + FA\u2026"), 1),
---------- 2022-04-12 12:16:20 -----------
```

Looking at the stdout, it appears to be working properly. I kill the consumer afterwards by using "yarn application -kill application_ID".

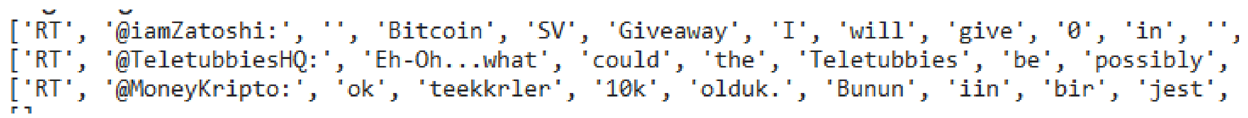
Final Question 2A Code here (final output on page 17)



kafka_consumer.py 4 ×

C: > Users > Eddie > Documents > School > CUHK > 2021 - 2022 Term 2 > IEMS 5730 - Big Data Systems and Information Processing > Week 13 > Assignment 4 > kafka_consumer.py > ...

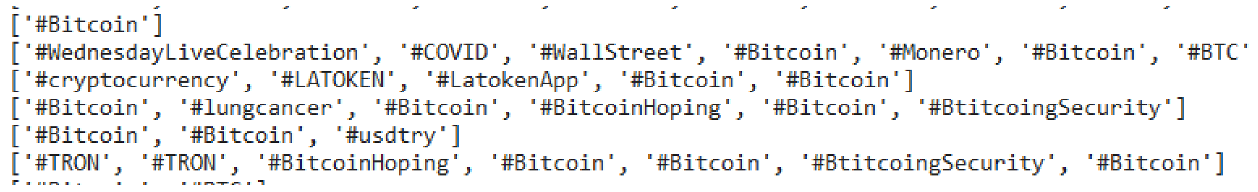
```python
from pyspark import SparkConf, SparkContext
from pyspark.streaming import StreamingContext
from pyspark.sql import Row, SQLContext
import sys
import json
import time
import pprint

if __name__ == '__main__':
    sc = SparkContext(appName="A4-Consumer") # Create spark context and set app name
    sc.setLogLevel("WARN") # set log level to warn

    ssc = StreamingContext(sc, 10) # Create streaming context with batch duration of 10 seconds
    ssc.checkpoint('./checkpoint') # Create checkpoint directory for fault tolerance.

    # Create Kafka stream (Streaming context, zookeeper, consumer group, {topic, number of partitions to consume per topic})
    kafkaStream = KafkaUtils.createStream(ssc, 'dicvmd7.ie.cuhk.edu.hk:2181', 's1155160788-consumers', {'1155160788-test': 1})

    words = kafkaStream.map(lambda x: x[1]).flatMap(lambda x: x.split(" ")).map(lambda x: x.encode("ascii", "ignore")) # Split tweets into words.

    # Filter the list of words to only include those with # as the first character and are more than just #, in other words, hashtags
    hashtags = words.filter(lambda hash: len(hash) > 2 and '#' == hash[0])
    countedHashtags = hashtags.countByValueAndWindow(300, 120) #count the number of occurences of each hashtag, over 5 minute period, slide every 2 minutes
    sortedHashtags = countedHashtags.transform((lambda x: x.sortBy(lambda x:( -x[1])))) # Sort hashtags by frequency in descending order.
    sortedHashtags.pprint(30) # Print the top 30 results from each slide interval

    ssc.start() # Start streaming context.
    ssc.awaitTermination() # Await termination
```



```
['RT', '@iamZatoshi:', '', 'Bitcoin', 'SV', 'Giveaway', 'I', 'will', 'give', '0', 'in', '',
['RT', '@TeletubbiesHQ:', 'Eh-Oh...what', 'could', 'the', 'Teletubbies', 'be', 'possibly',
['RT', '@MoneyKripto:', 'ok', 'teekkrler', '10k', 'olduk.', 'Bunun', 'iin', 'bir', 'jest',
```

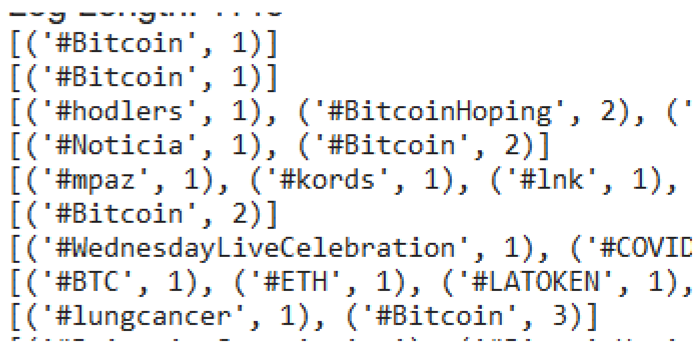
I get the code for splitting the tweet into words by combining the word count code from [16] with the original map code from [8]. I initially tried to keep the ['text'] part next to the x[1] but received an error saying that string indices needed to be integers, so I got rid of it and it still worked. There was originally a u' at the start of every word, which I learned from [18] meant that the string was a Unicode string. I used code from [19] to take the result from the flatMap and encode it as ascii so that the output looks normal like above.

To find hashtags, I filter the list of words (which includes all the hashtags), such that the first character of the word is #, and the length is greater than 2 using [19] as a brief inspiration. You might think "Why greater than 2 and not greater than 1?". I did examine the output of a test where the length was set to be greater than one and encountered a number of results like #t and #s that were at the end of the Tweet. The tweets are truncated and not showing the full tweet using ..., and often, it is split after the first letter of a hashtag, so I wanted to eliminate these results.



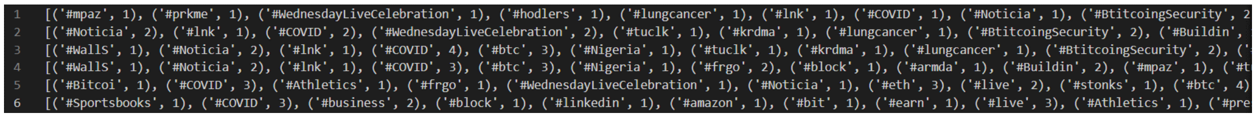
```
['#Bitcoin']
['#WednesdayLiveCelebration', '#COVID', '#WallStreet', '#Bitcoin', '#Monero', '#Bitcoin', '#BTC'
['#cryptocurrency', '#LATOKEN', '#LatokenApp', '#Bitcoin', '#Bitcoin']
['#Bitcoin', '#lungcancer', '#Bitcoin', '#BitcoinHoping', '#Bitcoin', '#BtitcoingSecurity']
['#Bitcoin', '#Bitcoin', '#usdtry']
['#TRON', '#TRON', '#BitcoinHoping', '#Bitcoin', '#Bitcoin', '#BtitcoingSecurity', '#Bitcoin']
```

I did validation with each hashtag to ensure that the list was accurate and that no legitimate hashtags were missed or being filtered out.



```
[('#Bitcoin', 1)]
[('#Bitcoin', 1)]
[('#hodlers', 1), ('#BitcoinHoping', 2), ('
[('#Noticia', 1), ('#Bitcoin', 2)]
[('#mpaz', 1), ('#kords', 1), ('#lnk', 1),
[('#Bitcoin', 2)]
[('#WednesdayLiveCelebration', 1), ('#COVID
[('#BTC', 1), ('#ETH', 1), ('#LATOKEN', 1),
[('#lungcancer', 1), ('#Bitcoin', 3)]
```

After applying countByValue, we have the number of each hashtag in each batch, which is 10 seconds long. According to the Spark 2.3.1 pyspark streaming API [20], we would want to use countByValueAndWindow instead of countByWindow. The later will only return a single value with the number of hashtags, while the former shows the counts of every distinct value. I set the parameters to 300 because 300 seconds = 5 minutes, while I use 120 for the sliding interval, because 120 seconds = 2 minutes, which is the sliding window we want. I test this for 12 minutes because depending on how it works, I will either get datapoints at 5,7,9, and 11 minutes or at 2, 4, 6, 8, 10, and 12 minutes.



```
[('#mpaz', 1), ('#prkme', 1), ('#WednesdayLiveCelebration', 1), ('#hodlers', 1), ('#lungcancer', 1), ('#lnk', 1), ('#COVID', 1), ('#Noticia', 1), ('#BtitcoingSecurity', 2
[('#Noticia', 2), ('#lnk', 1), ('#COVID', 2), ('#WednesdayLiveCelebration', 2), ('#tuclk', 1), ('#krdma', 1), ('#lungcancer', 1), ('#BtitcoingSecurity', 2), ('#Buildin',
[('#WallS', 1), ('#Noticia', 2), ('#lnk', 1), ('#COVID', 4), ('#btc', 3), ('#Nigeria', 1), ('#tuclk', 1), ('#krdma', 1), ('#lungcancer', 1), ('#BtitcoingSecurity', 2), ('
[('#WallS', 1), ('#Noticia', 2), ('#lnk', 1), ('#COVID', 3), ('#btc', 3), ('#Nigeria', 1), ('#frgo', 2), ('#block', 1), ('#armda', 1), ('#Buildin', 2), ('#mpaz', 1), ('#t
[('#Bitcoi', 1), ('#COVID', 3), ('#Athletics', 1), ('#frgo', 1), ('#WednesdayLiveCelebration', 1), ('#Noticia', 1), ('#eth', 3), ('#live', 2), ('#stonks', 1), ('#btc', 4)
[('#Sportsbooks', 1), ('#COVID', 3), ('#business', 2), ('#block', 1), ('#linkedin', 1), ('#amazon', 1), ('#bit', 1), ('#earn', 1), ('#live', 3), ('#Athletics', 1), ('#pre
```

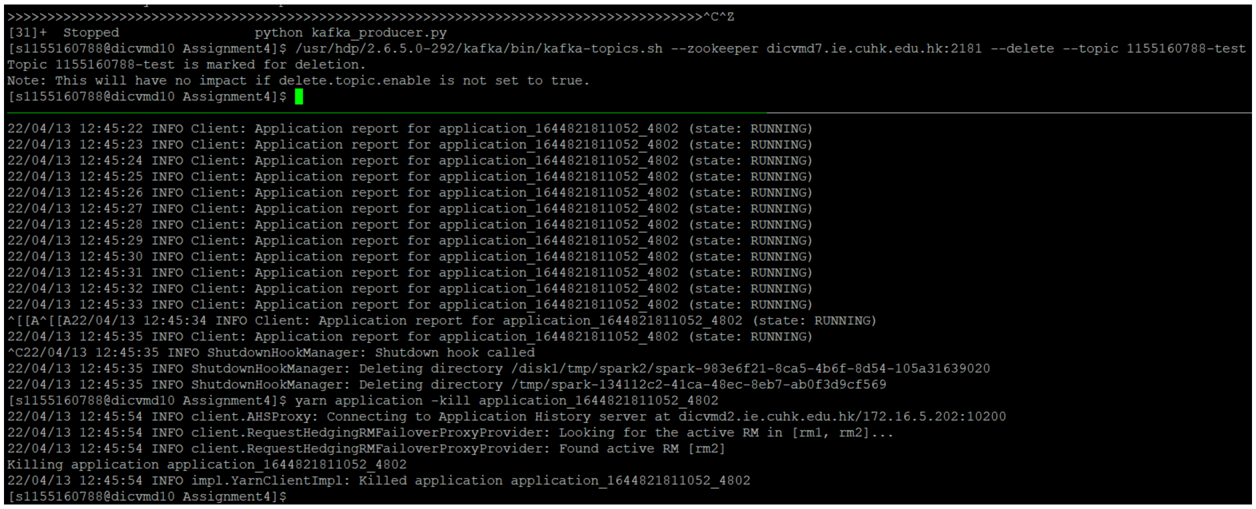
The results give me 6 lines that are very long horizontally, which means that when the slide interval is smaller than the slide window, it will process with whatever information it has, and it output a RDD / Dstream at 2, 4, 6, 8, 10, and 12 minutes. I checked to make sure that notable hashtags such as #Bitcoin were present in each sliding window, and the totals seemed reasonable. The length of the first line and the totals for each hashtag were reduced, but that is to be expected because it only had 2 minutes of data, while the 2nd line has 4 minutes of data, and the remaining lines have 5 minutes of data.

For sorting the results by frequency in descending order, I use the sortBy code from [16]. They note that if you just try to run sortBy on the DStream, it will not work because DStream has no built in sortBy method, so we need to instead transform it. -x[1] is used to sort in descending order.

I ran a test for 20 minutes to confirm the sort function worked, and the numbers looked reasonable. I also wanted to make sure the first sliding interval that only operated on 2 minutes of data had at least 30 hashtags, so there wouldn't be any errors from trying to take more than existed, and there was 34 distinct hashtags within the first 2 minutes, so that is not an issue.

I import and use pprint to print the top 30 hashtag results for each slide interval. As a result, I didn't need to use foreachRDD(), so I got rid of the process_rdd function.

Final submission for Question 2A:



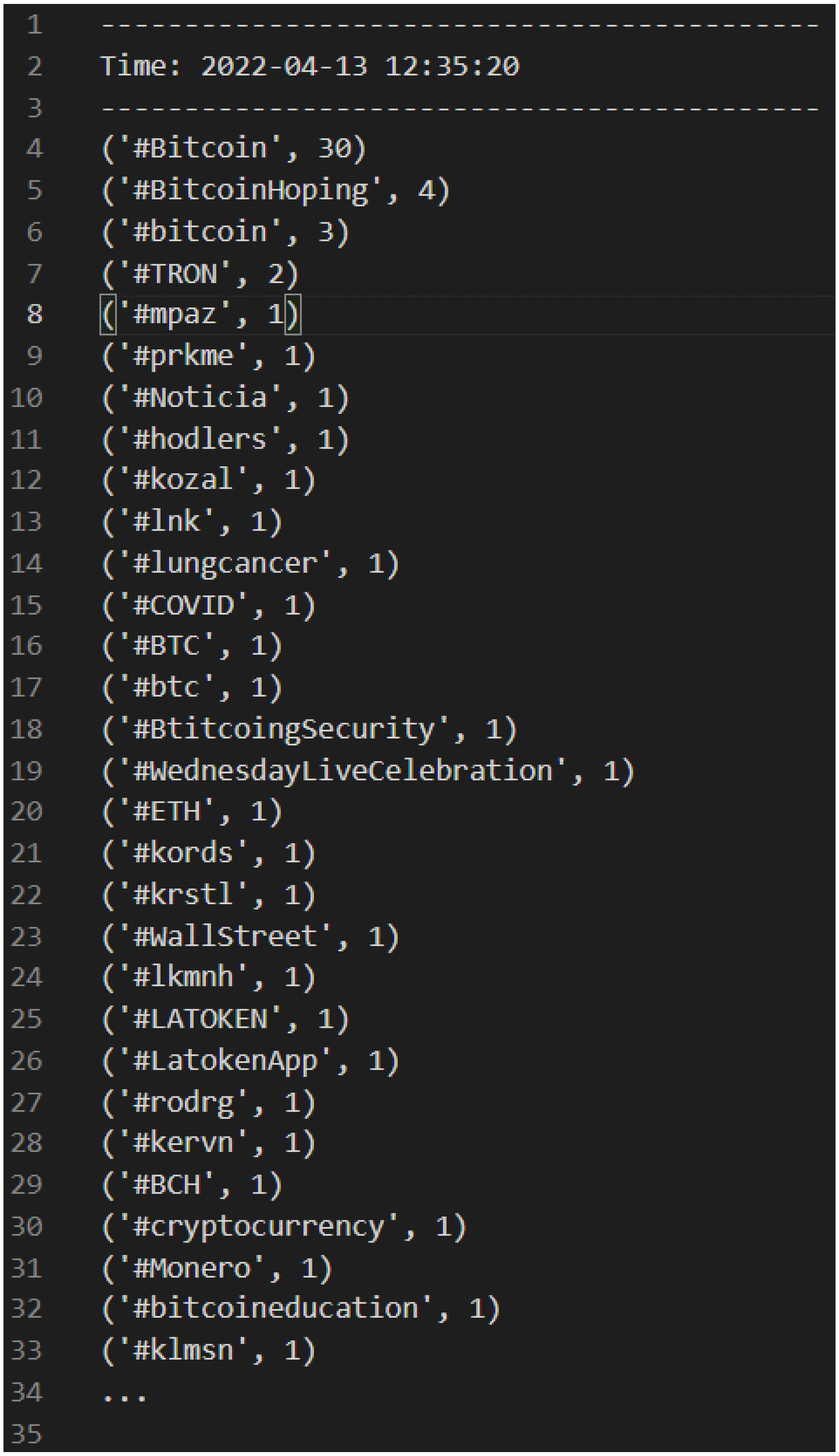
```
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>^C^Z
[31]+  Stopped                 python kafka_producer.py
[s1155160788@dicvmd10 Assignment4]$ /usr/hdp/2.6.5.0-292/kafka/bin/kafka-topics.sh --zookeeper dicvmd7.ie.cuhk.edu.hk:2181 --delete --topic 1155160788-test
Topic 1155160788-test is marked for deletion.
Note: This will have no impact if delete.topic.enable is not set to true.
[s1155160788@dicvmd10 Assignment4]$ 
```

```
22/04/13 12:45:22 INFO Client: Application report for application_1644821811052_4802 (state: RUNNING)
22/04/13 12:45:23 INFO Client: Application report for application_1644821811052_4802 (state: RUNNING)
22/04/13 12:45:24 INFO Client: Application report for application_1644821811052_4802 (state: RUNNING)
22/04/13 12:45:25 INFO Client: Application report for application_1644821811052_4802 (state: RUNNING)
22/04/13 12:45:26 INFO Client: Application report for application_1644821811052_4802 (state: RUNNING)
22/04/13 12:45:27 INFO Client: Application report for application_1644821811052_4802 (state: RUNNING)
22/04/13 12:45:28 INFO Client: Application report for application_1644821811052_4802 (state: RUNNING)
22/04/13 12:45:29 INFO Client: Application report for application_1644821811052_4802 (state: RUNNING)
22/04/13 12:45:30 INFO Client: Application report for application_1644821811052_4802 (state: RUNNING)
22/04/13 12:45:31 INFO Client: Application report for application_1644821811052_4802 (state: RUNNING)
22/04/13 12:45:32 INFO Client: Application report for application_1644821811052_4802 (state: RUNNING)
22/04/13 12:45:33 INFO Client: Application report for application_1644821811052_4802 (state: RUNNING)
^[[A^[[A22/04/13 12:45:34 INFO Client: Application report for application_1644821811052_4802 (state: RUNNING)
22/04/13 12:45:35 INFO Client: Application report for application_1644821811052_4802 (state: RUNNING)
^C22/04/13 12:45:35 INFO ShutdownHookManager: Shutdown hook called
22/04/13 12:45:35 INFO ShutdownHookManager: Deleting directory /disk1/tmp/spark2/spark-983e6f21-8ca5-4b6f-8d54-105a31639020
22/04/13 12:45:35 INFO ShutdownHookManager: Deleting directory /tmp/spark-134112c2-41ca-48ec-8eb7-ab0f3d9cf569
[s1155160788@dicvmd10 Assignment4]$ yarn application -kill application_1644821811052_4802
22/04/13 12:45:54 INFO client.AHSProxy: Connecting to Application History server at dicvmd2.ie.cuhk.edu.hk/172.16.5.202:10200
22/04/13 12:45:54 INFO client.RequestHedgingRMFailoverProxyProvider: Looking for the active RM in [rm1, rm2]...
22/04/13 12:45:54 INFO client.RequestHedgingRMFailoverProxyProvider: Found active RM [rm2]
Killing application application_1644821811052_4802
22/04/13 12:45:54 INFO impl.YarnClientImpl: Killed application application_1644821811052_4802
[s1155160788@dicvmd10 Assignment4]$
```

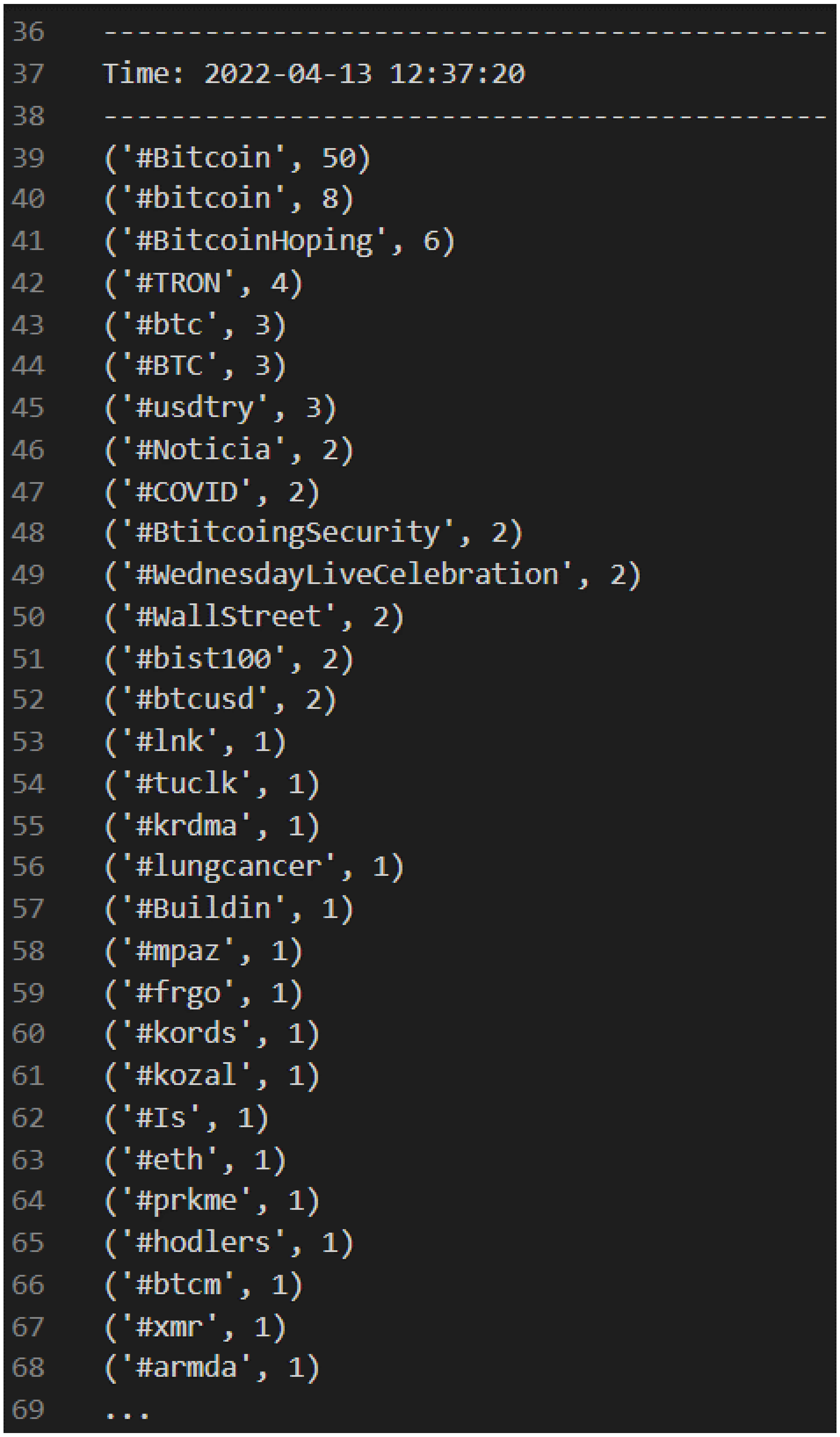
I will also upload this txt file onto Blackboard. I extracted it from the stdout log of the spark consumer program.

Because I use pprint (pretty print), it has a particular formatting. "..." means that the total list of hashtags is truncated, since it is only showing the top 30 results. The final results are with 12 minutes of data, and as such there is 6 RDD's resulting from 6 sliding intervals. You can see from the times listed that each one is 2 minutes apart, and based on lesser hashtag results for the first two sliding intervals (especially the first), the window is 5 minutes. Starting on the 3 slide interval (at 6 minutes), the hashtag counts roughly stabilize for the rest of execution.

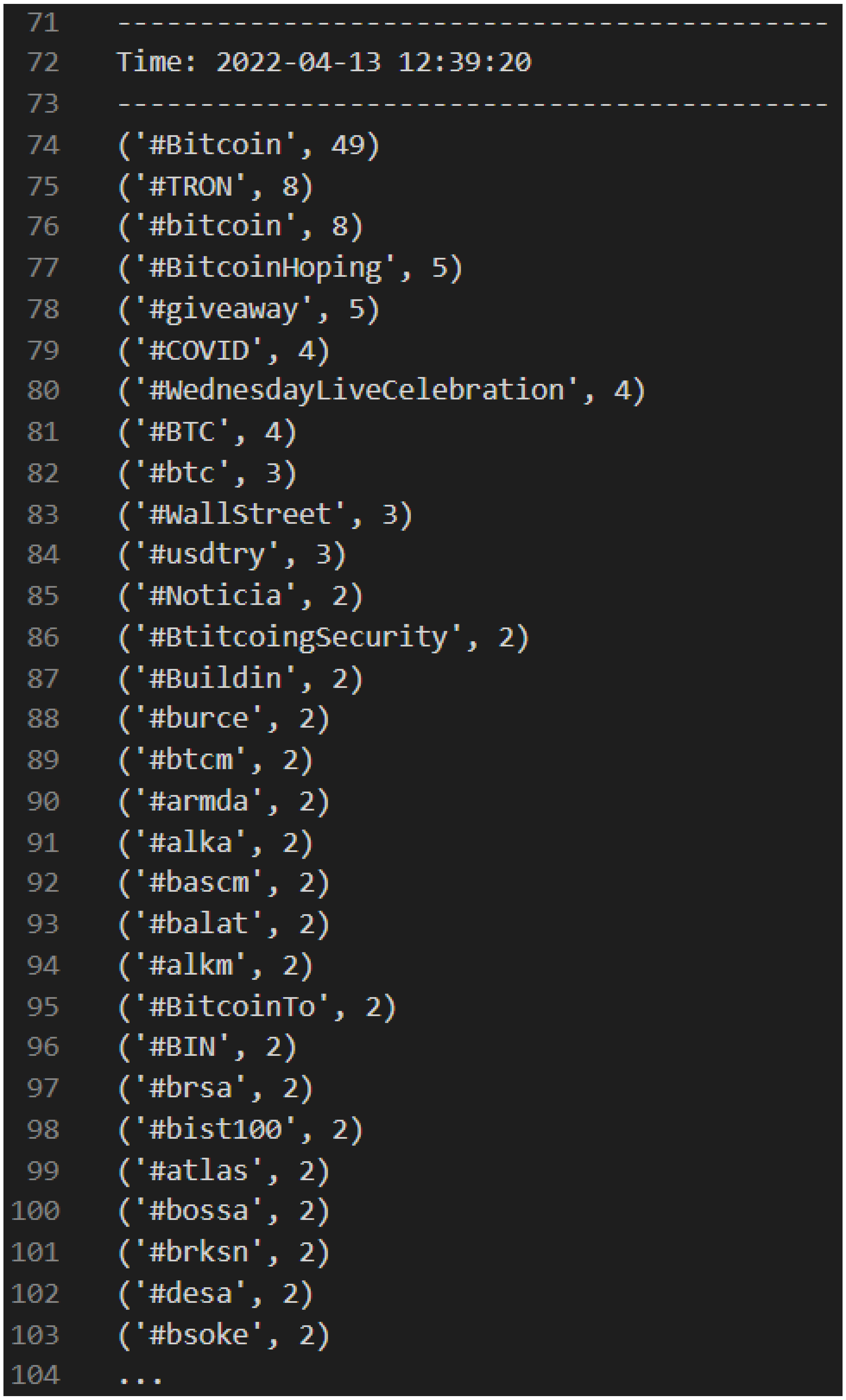
Screenshots start on next page, 6 total, 1 for each sliding interval.



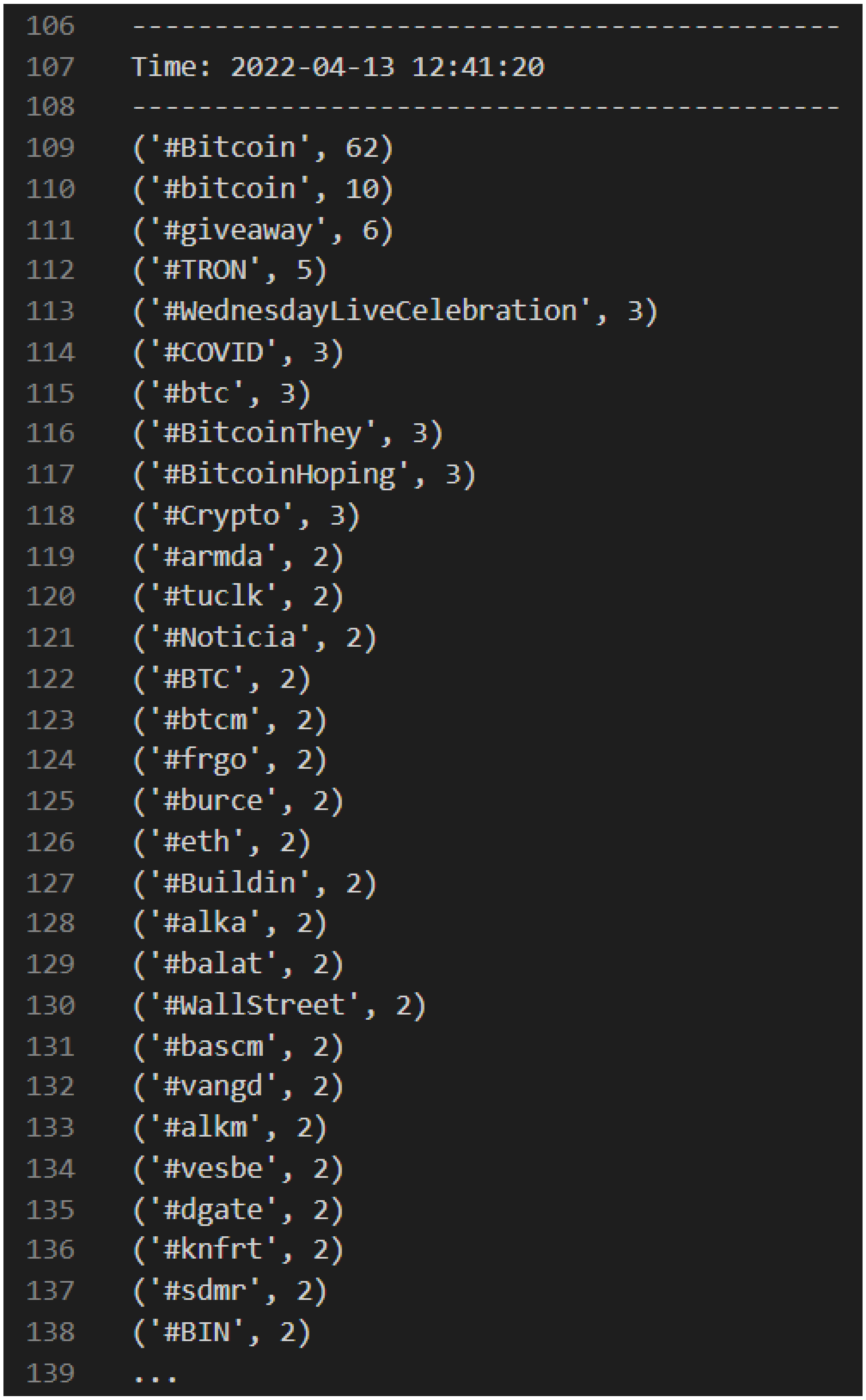
```
-------------------------------------------
Time: 2022-04-13 12:35:20
-------------------------------------------
('#Bitcoin', 30)
('#BitcoinHoping', 4)
('#bitcoin', 3)
('#TRON', 2)
('#mpaz', 1)
('#prkme', 1)
('#Noticia', 1)
('#hodlers', 1)
('#kozal', 1)
('#lnk', 1)
('#lungcancer', 1)
('#COVID', 1)
('#BTC', 1)
('#btc', 1)
('#BtitcoingSecurity', 1)
('#WednesdayLiveCelebration', 1)
('#ETH', 1)
('#kords', 1)
('#krstl', 1)
('#WallStreet', 1)
('#lkmnh', 1)
('#LATOKEN', 1)
('#LatokenApp', 1)
('#rodrg', 1)
('#kervn', 1)
('#BCH', 1)
('#cryptocurrency', 1)
('#Monero', 1)
('#bitcoineducation', 1)
('#klmsn', 1)
...
```



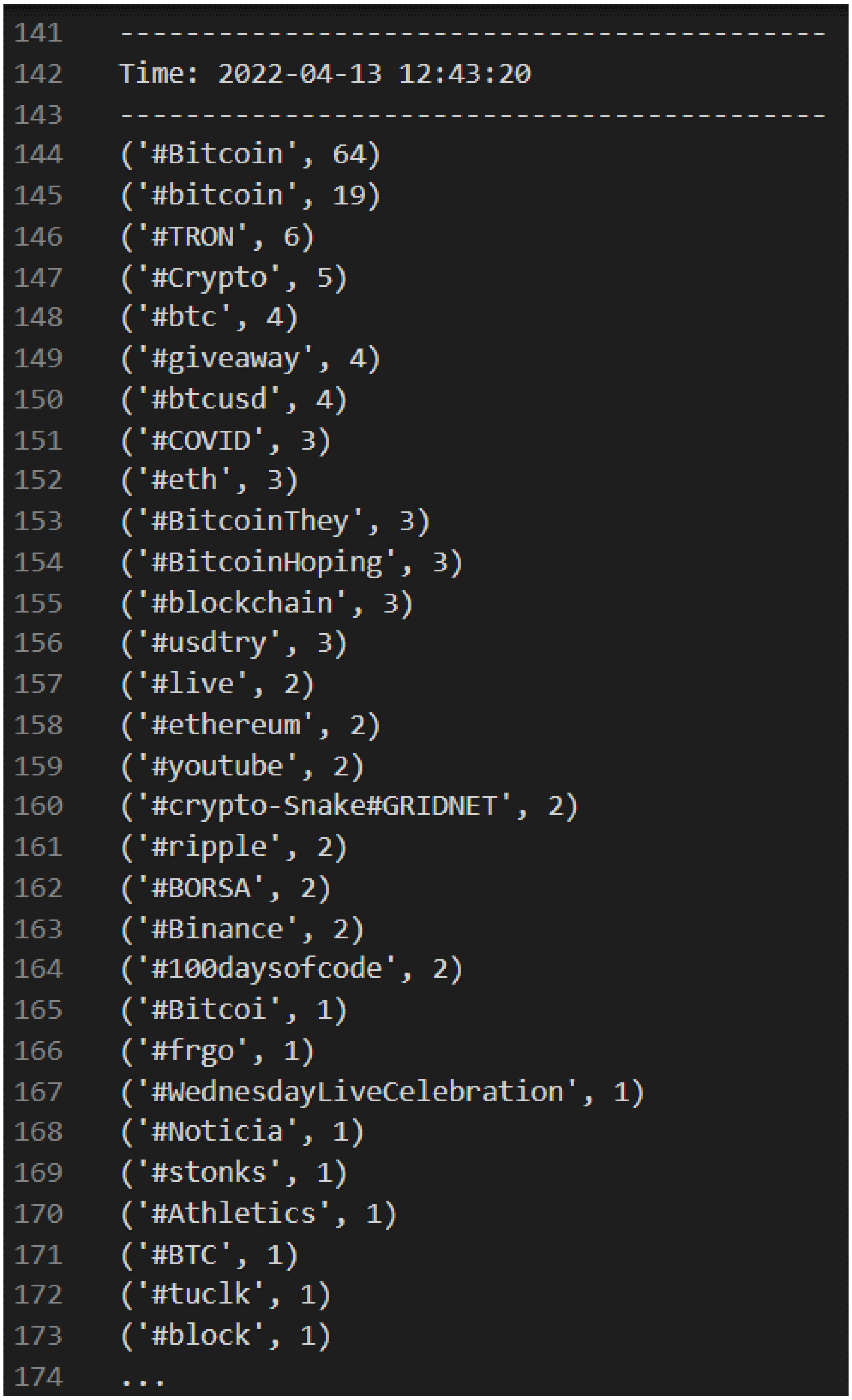
```
-------------------------------------------
Time: 2022-04-13 12:37:20
-------------------------------------------
('#Bitcoin', 50)
('#bitcoin', 8)
('#BitcoinHoping', 6)
('#TRON', 4)
('#btc', 3)
('#BTC', 3)
('#usdtry', 3)
('#Noticia', 2)
('#COVID', 2)
('#BtitcoingSecurity', 2)
('#WednesdayLiveCelebration', 2)
('#WallStreet', 2)
('#bist100', 2)
('#btcusd', 2)
('#lnk', 1)
('#tuclk', 1)
('#krdma', 1)
('#lungcancer', 1)
('#Buildin', 1)
('#mpaz', 1)
('#frgo', 1)
('#kords', 1)
('#kozal', 1)
('#Is', 1)
('#eth', 1)
('#prkme', 1)
('#hodlers', 1)
('#btcm', 1)
('#xmr', 1)
('#armda', 1)
...
```



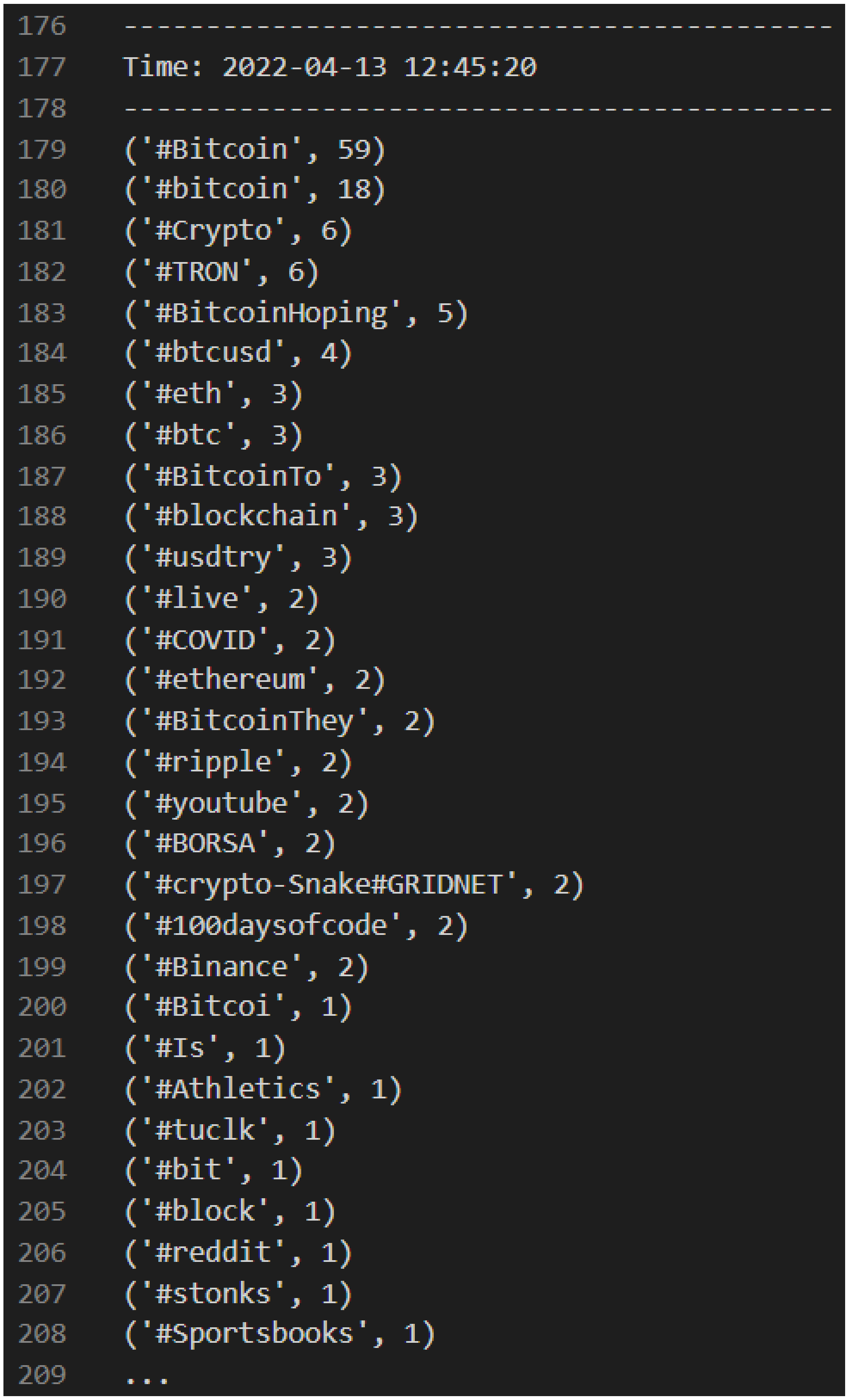
```
-------------------------------------------
Time: 2022-04-13 12:39:20
-------------------------------------------
('#Bitcoin', 49)
('#TRON', 8)
('#bitcoin', 8)
('#BitcoinHoping', 5)
('#giveaway', 5)
('#COVID', 4)
('#WednesdayLiveCelebration', 4)
('#BTC', 4)
('#btc', 3)
('#WallStreet', 3)
('#usdtry', 3)
('#Noticia', 2)
('#BtitcoingSecurity', 2)
('#Buildin', 2)
('#burce', 2)
('#btcm', 2)
('#armda', 2)
('#alka', 2)
('#bascm', 2)
('#balat', 2)
('#alkm', 2)
('#BitcoinTo', 2)
('#BIN', 2)
('#brsa', 2)
('#bist100', 2)
('#atlas', 2)
('#bossa', 2)
('#brksn', 2)
('#desa', 2)
('#bsoke', 2)
...
```



```
-------------------------------------------
Time: 2022-04-13 12:41:20
-------------------------------------------
('#Bitcoin', 62)
('#bitcoin', 10)
('#giveaway', 6)
('#TRON', 5)
('#WednesdayLiveCelebration', 3)
('#COVID', 3)
('#btc', 3)
('#BitcoinThey', 3)
('#BitcoinHoping', 3)
('#Crypto', 3)
('#armda', 2)
('#tuclk', 2)
('#Noticia', 2)
('#BTC', 2)
('#btcm', 2)
('#frgo', 2)
('#burce', 2)
('#eth', 2)
('#Buildin', 2)
('#alka', 2)
('#balat', 2)
('#WallStreet', 2)
('#bascm', 2)
('#vangd', 2)
('#alkm', 2)
('#vesbe', 2)
('#dgate', 2)
('#knfrt', 2)
('#sdmr', 2)
('#BIN', 2)
...
```



```
-------------------------------------------
Time: 2022-04-13 12:43:20
-------------------------------------------
('#Bitcoin', 64)
('#bitcoin', 19)
('#TRON', 6)
('#Crypto', 5)
('#btc', 4)
('#giveaway', 4)
('#btcusd', 4)
('#COVID', 3)
('#eth', 3)
('#BitcoinThey', 3)
('#BitcoinHoping', 3)
('#blockchain', 3)
('#usdtry', 3)
('#live', 2)
('#ethereum', 2)
('#youtube', 2)
('#crypto-Snake#GRIDNET', 2)
('#ripple', 2)
('#BORSA', 2)
('#Binance', 2)
('#100daysofcode', 2)
('#Bitcoi', 1)
('#frgo', 1)
('#WednesdayLiveCelebration', 1)
('#Noticia', 1)
('#stonks', 1)
('#Athletics', 1)
('#BTC', 1)
('#tuclk', 1)
('#block', 1)
...
```

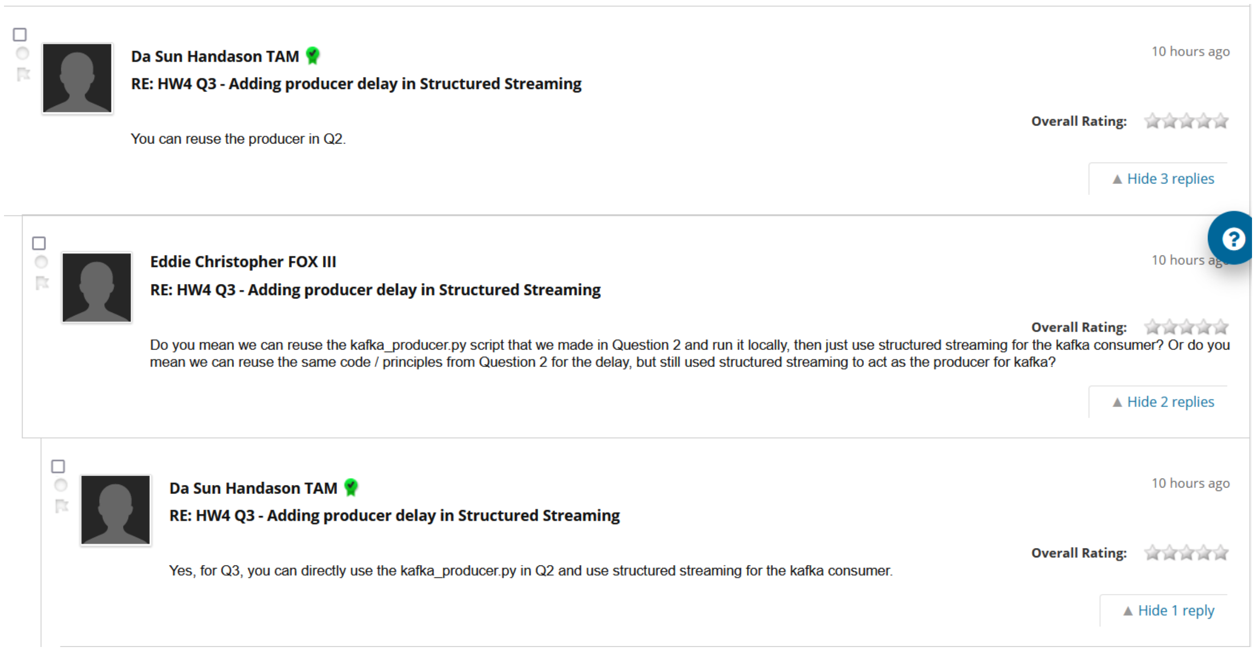


```
-------------------------------------------
Time: 2022-04-13 12:45:20
-------------------------------------------
('#Bitcoin', 59)
('#bitcoin', 18)
('#Crypto', 6)
('#TRON', 6)
('#BitcoinHoping', 5)
('#btcusd', 4)
('#eth', 3)
('#btc', 3)
('#BitcoinTo', 3)
('#blockchain', 3)
('#usdtry', 3)
('#live', 2)
('#COVID', 2)
('#ethereum', 2)
('#BitcoinThey', 2)
('#ripple', 2)
('#youtube', 2)
('#BORSA', 2)
('#crypto-Snake#GRIDNET', 2)
('#100daysofcode', 2)
('#Binance', 2)
('#Bitcoi', 1)
('#Is', 1)
('#Athletics', 1)
('#tuclk', 1)
('#bit', 1)
('#block', 1)
('#reddit', 1)
('#stonks', 1)
('#Sportsbooks', 1)
...
```

Question 3: Spark Structured Streaming

(Final submission on page 29)

Note: I spent a very long time trying to integrate structured streaming for both the producer and consumer into the same Scala Spark program and also trying to implement the artificial timestamp delay in Scala / Structured Streaming. This was because there is resource limitations preventing us from running two Spark jobs simultaneously, and I thought that I was not allowed to reuse the python kafka producer from Question 2. I will not document all my attempts to make things work under this arrangement as I encountered lots of errors.


Da Sun Handason TAM
RE: HW4 Q3 - Adding producer delay in Structured Streaming
10 hours ago
Overall Rating:

You can reuse the producer in Q2.

▲ Hide 3 replies

Eddie Christopher FOX III
RE: HW4 Q3 - Adding producer delay in Structured Streaming
10 hours ago
Overall Rating:

Do you mean we can reuse the kafka_producer.py script that we made in Question 2 and run it locally, then just use structured streaming for the kafka consumer? Or do you mean we can reuse the same code / principles from Question 2 for the delay, but still used structured streaming to act as the producer for kafka?

▲ Hide 2 replies

Da Sun Handason TAM
RE: HW4 Q3 - Adding producer delay in Structured Streaming
10 hours ago
Overall Rating:

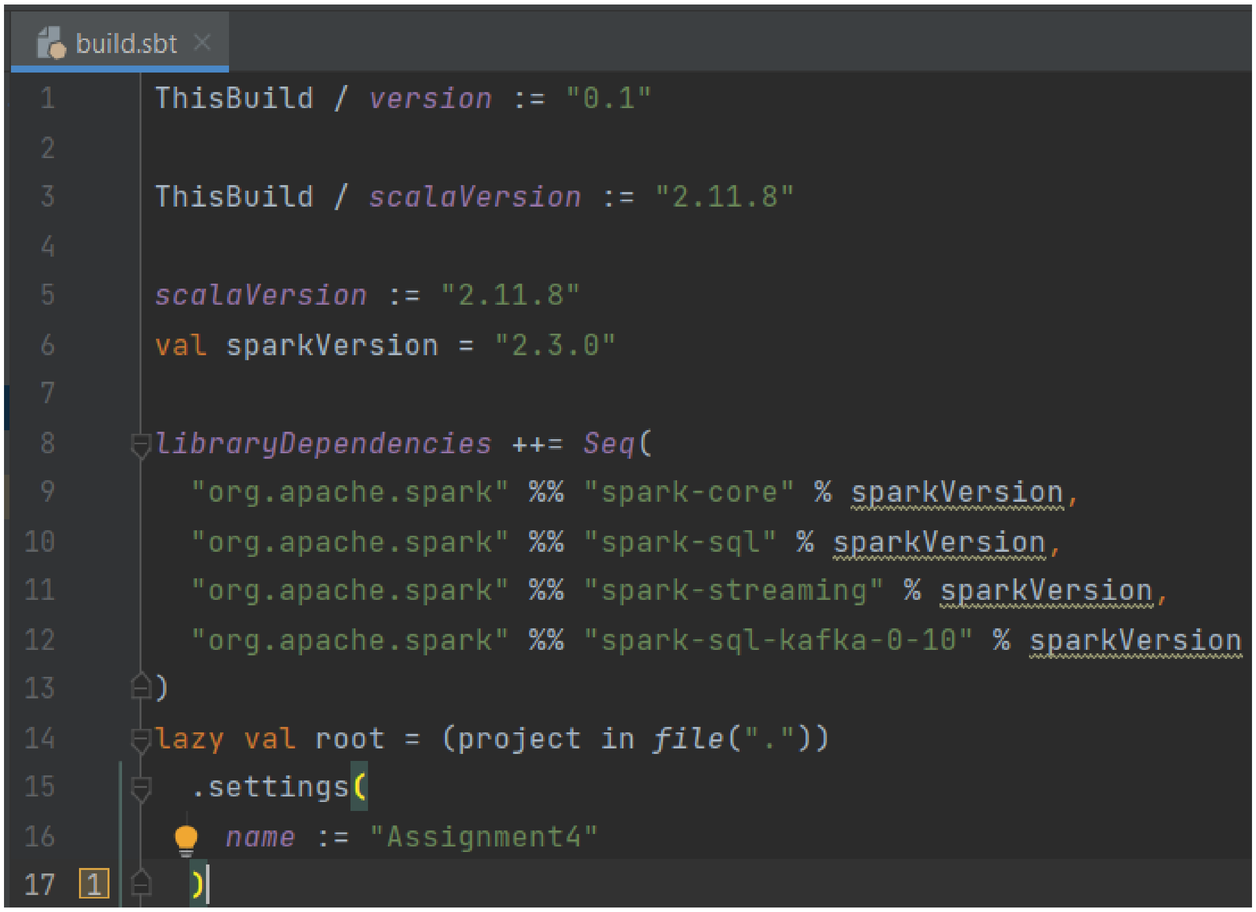
Yes, for Q3, you can directly use the kafka_producer.py in Q2 and use structured streaming for the kafka consumer.

▲ Hide 1 reply

I decide to write this in Scala because I have experience writing Spark dataframes in Scala from Assignment 3. I will reuse the kafka_producer.py from question 2 to take advantage of what has already been created.

I created a new project in Intellij for this. I refer to [21], [22], and [23] while programming the structured streaming.

Screenshot on next page.



```
ThisBuild / version := "0.1"

ThisBuild / scalaVersion := "2.11.8"

scalaVersion := "2.11.8"
val sparkVersion = "2.3.0"

libraryDependencies ++= Seq(
  "org.apache.spark" %% "spark-core" % sparkVersion,
  "org.apache.spark" %% "spark-sql" % sparkVersion,
  "org.apache.spark" %% "spark-streaming" % sparkVersion,
  "org.apache.spark" %% "spark-sql-kafka-0-10" % sparkVersion
)
lazy val root = (project in file("."))
  .settings(
    name := "Assignment4"
  )
```

build.sbt file here. I mostly copied from my Assignment 3 build.sbt file but added the spark-sql-kafka-0-10 dependency for structured streaming, and changed the assignment name. I'm not sure if I need spark-streaming but decided to add it just in case.

I was initially receiving errors when trying spark-sql-kafka-0-10_2.11, but resolved this by getting rid of the _2.11 as suggested in [24].

I create a Question3 package under /src/main/scala and a KafkaStreams object class.

I start with the same import statements, def main, and spark session builder statements as Assignment 3. This is also mentioned in [22].

I navigate to the proper directory and run "sbt package" to compile the jar file, then upload the jar file to the IE DIC cluster and HDFS.

Spark submission command:

```
"spark-submit \
        --packages org.apache.spark:spark-sql-kafka-0-10_2.11:2.3.0 \
        --class Question3.KafkaStreams \
        --master yarn \
        --deploy-mode cluster \
```

hdfs://dicvmd2.ie.cuhk.edu.hk:8020/user/s1155160788/assignment4_2.11-0.1.jar"

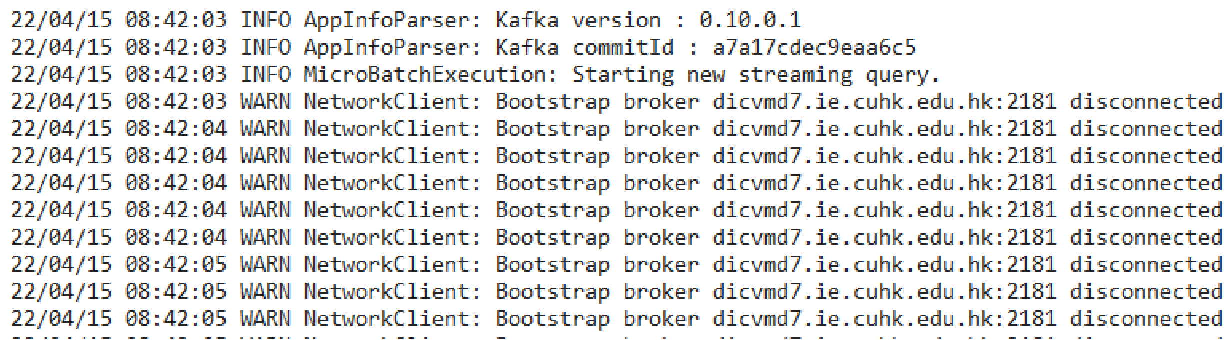
I had an error earlier where it said "java.lang.ClassNotFoundException: Failed to find data source: kafka. Please find packages at http://spark.apache.org/third-party-projects.html" but I remedied this by adding "--packages org.apache.spark:spark-sql-kafka-0-10_2.11:2.3.0" to the spark-submit command [8] [25].

I refer to the Kafka Integration Guide[21] to write the .readStream command and to [26] for writing the writeStream file sink.

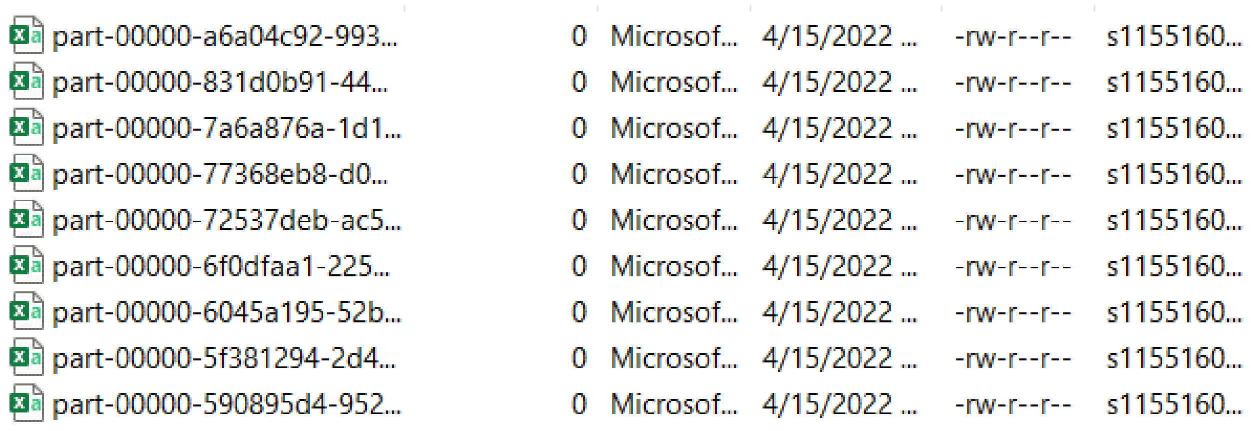
When doing some initial testing, I ran into the error: "User class threw exception: org.apache.spark.sql.AnalysisException: Try to map struct<key:binary,value:binary,topic:string,partition:int,offset:bigint,timestamp:timestamp,timestampType:int> to Tuple1, but failed as the number of fields does not line up.;"

To fix this, as suggested in [27], I changed to "consumer = consumer .selectExpr("CAST(key AS STRING)", "CAST(value AS STRING)")" to "val lines = consumer.selectExpr("CAST(value AS STRING)").as[String]". I did not need the key, and also doing consumer.selectExpr would not actually transform consumer since it was immutable.

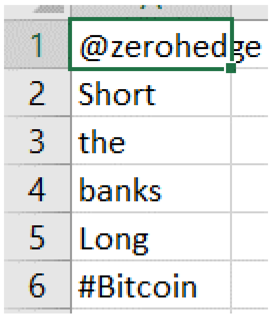
For a long time, when trying to test, I would get an empty output directory with only a _spark_metadata folder with nothing inside it. After some inspiration from [28], I checked the logs, and discovered that it continually said that the bootstrap broker was disconnected after the streaming query started.


```
22/04/15 08:42:03 INFO AppInfoParser: Kafka version : 0.10.0.1
22/04/15 08:42:03 INFO AppInfoParser: Kafka commitId : a7a17cdec9eaa6c5
22/04/15 08:42:03 INFO MicroBatchExecution: Starting new streaming query.
22/04/15 08:42:03 WARN NetworkClient: Bootstrap broker dicvmd7.ie.cuhk.edu.hk:2181 disconnected
22/04/15 08:42:04 WARN NetworkClient: Bootstrap broker dicvmd7.ie.cuhk.edu.hk:2181 disconnected
22/04/15 08:42:04 WARN NetworkClient: Bootstrap broker dicvmd7.ie.cuhk.edu.hk:2181 disconnected
22/04/15 08:42:04 WARN NetworkClient: Bootstrap broker dicvmd7.ie.cuhk.edu.hk:2181 disconnected
22/04/15 08:42:04 WARN NetworkClient: Bootstrap broker dicvmd7.ie.cuhk.edu.hk:2181 disconnected
22/04/15 08:42:04 WARN NetworkClient: Bootstrap broker dicvmd7.ie.cuhk.edu.hk:2181 disconnected
22/04/15 08:42:05 WARN NetworkClient: Bootstrap broker dicvmd7.ie.cuhk.edu.hk:2181 disconnected
22/04/15 08:42:05 WARN NetworkClient: Bootstrap broker dicvmd7.ie.cuhk.edu.hk:2181 disconnected
22/04/15 08:42:05 WARN NetworkClient: Bootstrap broker dicvmd7.ie.cuhk.edu.hk:2181 disconnected
```

Based on [28], Structured Streaming cannot accept a zookeeper server, so even through I was able to use port 2181 as the designated bootstrap in the kafka_consumer.py in Question 2, I needed to change the port to 6667 for question 3. After changing this parameter and recompiling the JAR file, I started getting output.



part-00000-a6a04c92-993...	0	Microsof...	4/15/2022 ...	-rw-r--r--	s1155160...
part-00000-831d0b91-44...	0	Microsof...	4/15/2022 ...	-rw-r--r--	s1155160...
part-00000-7a6a876a-1d1...	0	Microsof...	4/15/2022 ...	-rw-r--r--	s1155160...
part-00000-77368eb8-d0...	0	Microsof...	4/15/2022 ...	-rw-r--r--	s1155160...
part-00000-72537deb-ac5...	0	Microsof...	4/15/2022 ...	-rw-r--r--	s1155160...
part-00000-6f0dfaa1-225...	0	Microsof...	4/15/2022 ...	-rw-r--r--	s1155160...
part-00000-6045a195-52b...	0	Microsof...	4/15/2022 ...	-rw-r--r--	s1155160...
part-00000-5f381294-2d4...	0	Microsof...	4/15/2022 ...	-rw-r--r--	s1155160...
part-00000-590895d4-952...	0	Microsof...	4/15/2022 ...	-rw-r--r--	s1155160...



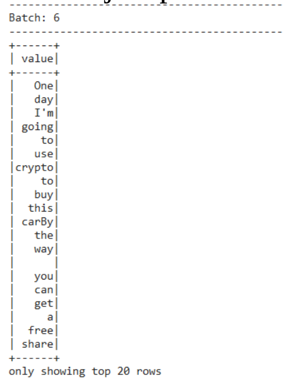
1	@zerohedge
2	Short
3	the
4	banks
5	Long
6	#Bitcoin

Many CSV files were being generated, so I changed it based on [29] to set the batch duration to 10 seconds so it would only generate one result / batch / CSV file every 10 seconds.

I received the following error during testing: "User class threw exception: org.apache.spark.sql.streaming.StreamingQueryException: Partition 1155160788-test-1's offset was changed from 54 to 15, some data may have been missed.

Some data may have been lost because they are not available in Kafka any more; either the data was aged out by Kafka or the topic may have been deleted before all the data in the topic was processed. If you don't want your streaming query to fail on such cases, set the source option "failOnDataLoss" to "false"."

I resolved this based on [30] by deleting and recreating my checkpoint directory in HDFS. It may have had something to do with me changing the output format from CSV to console, or maybe there is just periodic changes in the topic offsets.



```
-------------------------------------------
Batch: 6
-------------------------------------------
+------+
| value|
+------+
|   One|
|   day|
|   I'm|
| going|
|    to|
|   use|
|crypto|
|    to|
|   buy|
|  this|
| carBy|
|   the|
|   way|
|      |
|   you|
|   can|
|   get|
|     a|
|  free|
| share|
+------+
only showing top 20 rows
```

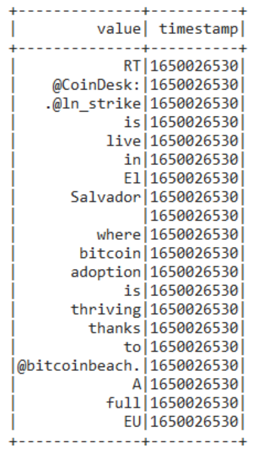
Example of batch output after just flatMap splitting into words from the console.

I refer to [31] for startsWith filter syntax on dataframes such that the word must start with #, and [32] for filtering so that the length is greater than 2, like in question 2.

The documentation for Spark Structured streaming says that to use window functions (with window duration and sliding interval), there needs to be a time based column[22].

I tried adding a column with the current timestamp using .withColumn("Timestamp", current_timestamp()) but it seems there is a bug on the version of Spark that the IE DIC cluster uses that prevents this with working [33], and was fixed in a later version, but that is not an option for this assignment.

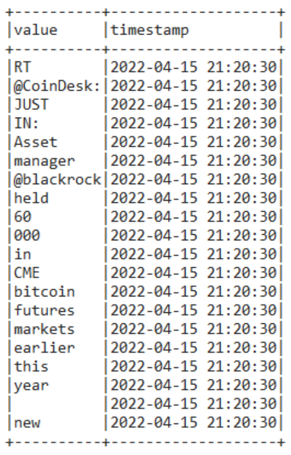
With no option to create a new timestamp column due to the Spark bugs, I am left with no choice but to use the timestamps provided from Kafka. This has the side effect of using the time the Tweets are ingested into Kafka rather than the time the datapoints were actually produced, but we are not doing timestamp based analysis, we only need it for windowed operation.
Using [21] and [34] as a reference, I create a select expression and cast the timestamp appropriately in order to include the timestamp with each Tweet. A further challenge comes from the flatMap operation I need to perform on the dataframe in order to flatMap only the value (text) column to split the words. I manage to do this with inspiration from [35] and guidance from [36] for the exact syntax, essentially turning the dataframe into an RDD, performing a case based flat map on just the value column, then converting it back into a dataframe.



```
+--------------+----------+
|         value| timestamp|
+--------------+----------+
|            RT|1650026530|
|    @CoinDesk:|1650026530|
|   .@ln_strike|1650026530|
|            is|1650026530|
|          live|1650026530|
|            in|1650026530|
|            El|1650026530|
|      Salvador|1650026530|
|              |1650026530|
|         where|1650026530|
|       bitcoin|1650026530|
|      adoption|1650026530|
|            is|1650026530|
|      thriving|1650026530|
|        thanks|1650026530|
|            to|1650026530|
|@bitcoinbeach.|1650026530|
|             A|1650026530|
|          full|1650026530|
|            EU|1650026530|
+--------------+----------+
```

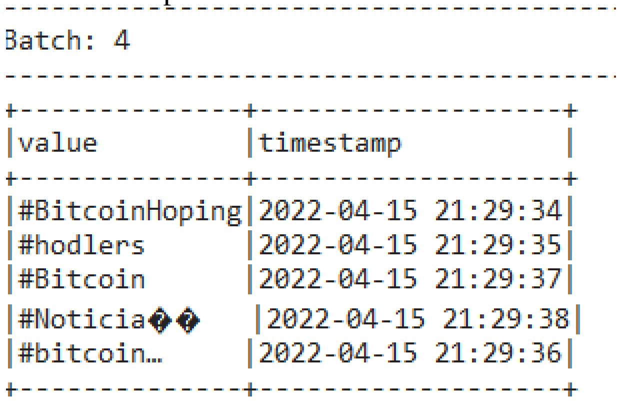
I get output like this from the Kafka timestamp, which is presumably each the number of seconds since 1970. The timestamps are all the same because each Tweet has the same associated kafka timestamp, but this is precise enough to enabled windowed operations with Structured Streaming.

I use [37] to convert the unix timestamp to a standard timestamp stream with from_unixtime(), and use the notation in [38] to update all values in the timestamp column with their timestamp equivalent. Finally, I use [39] to make sure they are in the timestamp data type by using to_timestamp() with the proper input format. The end result is output that looks like that on the next page.



```
+----------+-------------------+
|value     |timestamp          |
+----------+-------------------+
|RT        |2022-04-15 21:20:30|
|@CoinDesk:|2022-04-15 21:20:30|
|JUST      |2022-04-15 21:20:30|
|IN:       |2022-04-15 21:20:30|
|Asset     |2022-04-15 21:20:30|
|manager   |2022-04-15 21:20:30|
|@blackrock|2022-04-15 21:20:30|
|held      |2022-04-15 21:20:30|
|60        |2022-04-15 21:20:30|
|000       |2022-04-15 21:20:30|
|in        |2022-04-15 21:20:30|
|CME       |2022-04-15 21:20:30|
|bitcoin   |2022-04-15 21:20:30|
|futures   |2022-04-15 21:20:30|
|markets   |2022-04-15 21:20:30|
|earlier   |2022-04-15 21:20:30|
|this      |2022-04-15 21:20:30|
|year      |2022-04-15 21:20:30|
|          |2022-04-15 21:20:30|
|new       |2022-04-15 21:20:30|
+----------+-------------------+
```

I implement my code back to where I have the hashtags filtered and need to perform the windowed operations.



```
-------------------------------------------
Batch: 4
-------------------------------------------
+--------------+-------------------+
|value         |timestamp          |
+--------------+-------------------+
|#BitcoinHoping|2022-04-15 21:29:34|
|#hodlers      |2022-04-15 21:29:35|
|#Bitcoin      |2022-04-15 21:29:37|
|#Noticia��    |2022-04-15 21:29:38|
|#bitcoin…     |2022-04-15 21:29:36|
+--------------+-------------------+
```

We are starting to see much more different timestamps because generally there is not more than 1 or 2 hashtags per tweet, and each tweet has a different timestamp.

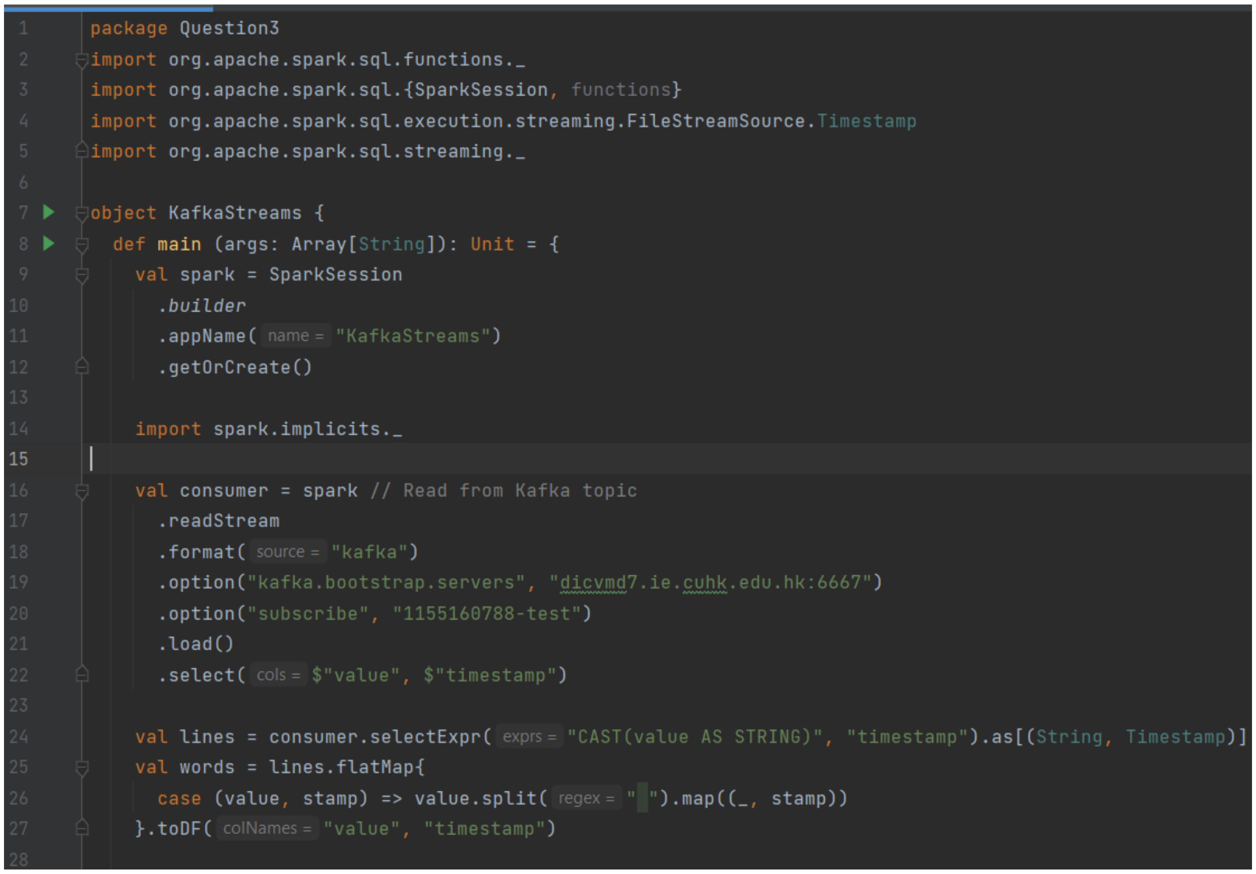
I referred to [22] and [40] for advice on how to perform the window group by operation and to [41] to understand the windowed operation options better, as well as [42] for official github sample code from the Apache documentation.

Question 3 Final Submission:

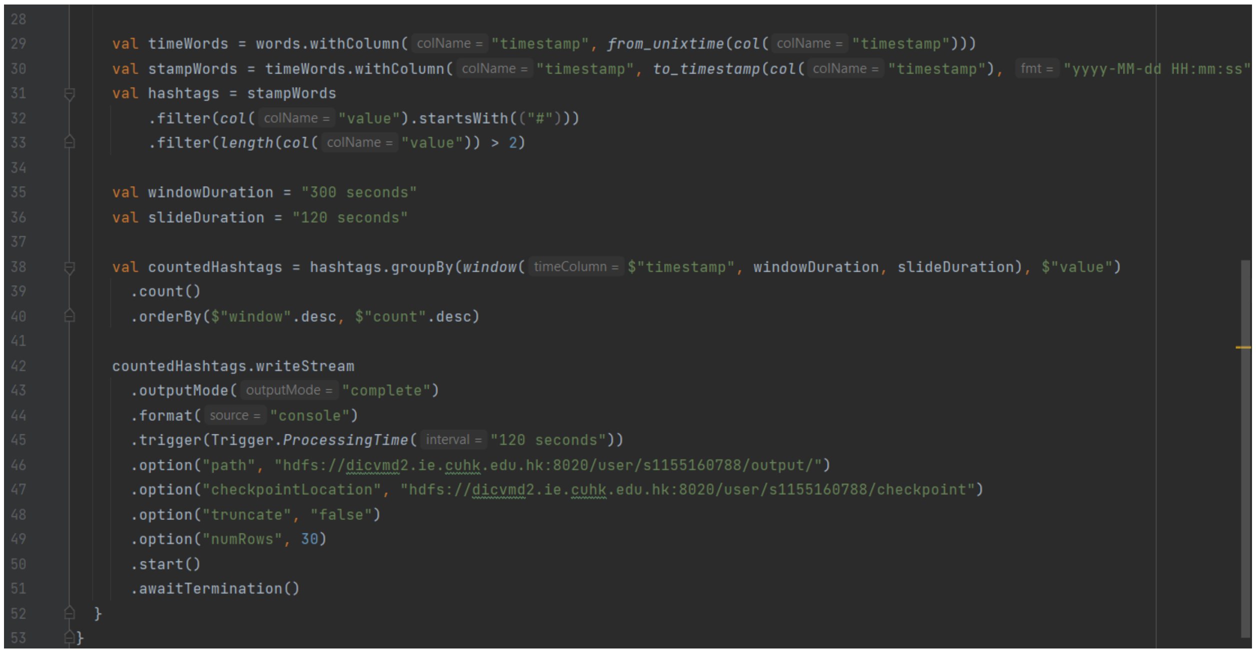
Final Spark submission command:

```
“spark-submit \
        --packages org.apache.spark:spark-sql-kafka-0-10_2.11:2.3.0 \
        --class Question3.KafkaStreams \
        --master yarn \
        --deploy-mode cluster \
        hdfs://dicvmd2.ie.cuhk.edu.hk:8020/user/s1155160788/assignment4_2.11-0.1.jar”
```

Final Spark Structured Streaming consumer code (Reused kafka_producer.py from question 2):

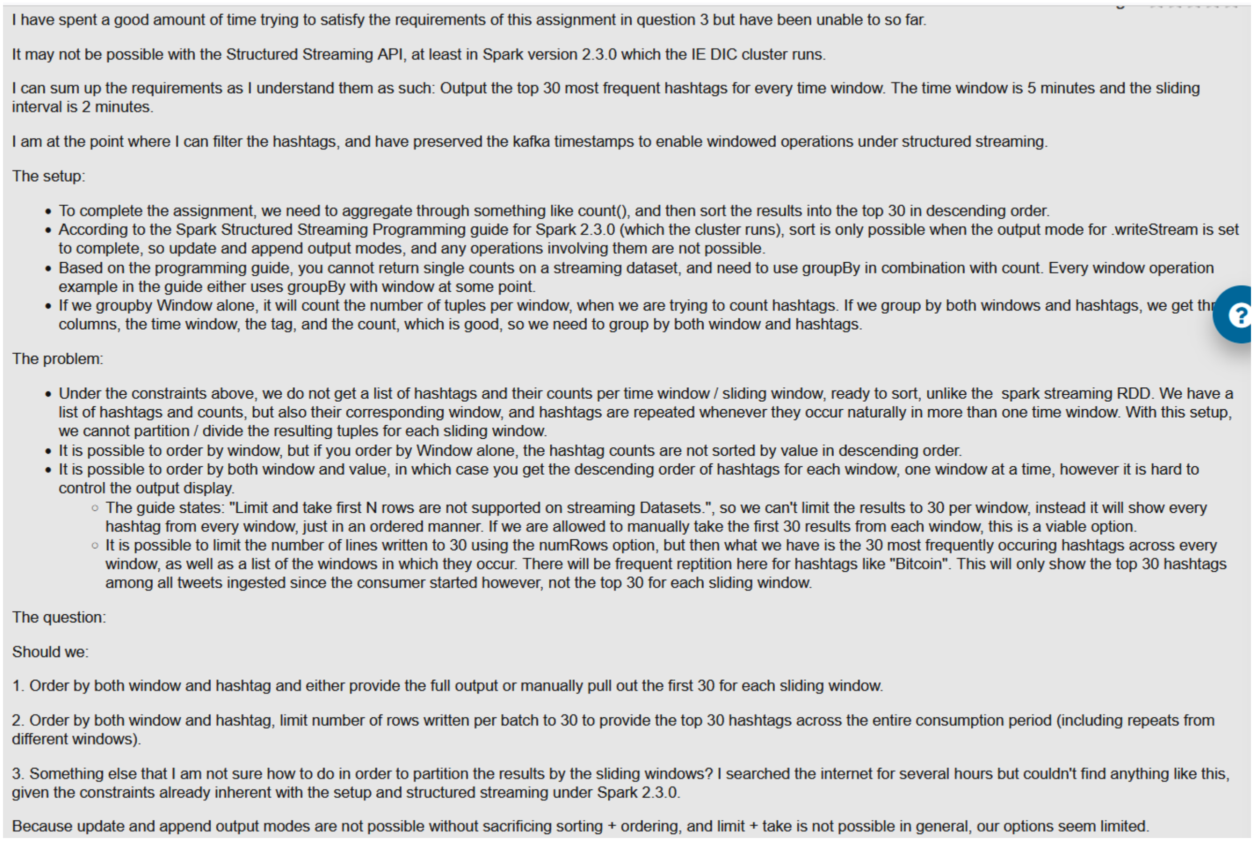


```scala
package Question3
import org.apache.spark.sql.functions._
import org.apache.spark.sql.{SparkSession, functions}
import org.apache.spark.sql.execution.streaming.FileStreamSource.Timestamp
import org.apache.spark.sql.streaming._

object KafkaStreams {
  def main (args: Array[String]): Unit = {
    val spark = SparkSession
      .builder
      .appName("KafkaStreams")
      .getOrCreate()

    import spark.implicits._

    val consumer = spark // Read from Kafka topic
      .readStream
      .format("kafka")
      .option("kafka.bootstrap.servers", "dicvmd7.ie.cuhk.edu.hk:6667")
      .option("subscribe", "1155160788-test")
      .load()
      .select($"value", $"timestamp")

    val lines = consumer.selectExpr("CAST(value AS STRING)", "timestamp").as[(String, Timestamp)]
    val words = lines.flatMap{
      case (value, stamp) => value.split(" ").map((_, stamp))
    }.toDF("value", "timestamp")
```



```scala
    val timeWords = words.withColumn("timestamp", from_unixtime(col("timestamp")))
    val stampWords = timeWords.withColumn("timestamp", to_timestamp(col("timestamp"), "yyyy-MM-dd HH:mm:ss"
    val hashtags = stampWords
        .filter(col("value").startsWith(("#")))
        .filter(length(col("value")) > 2)

    val windowDuration = "300 seconds"
    val slideDuration = "120 seconds"

    val countedHashtags = hashtags.groupBy(window($"timestamp", windowDuration, slideDuration), $"value")
      .count()
      .orderBy($"window".desc, $"count".desc)

    countedHashtags.writeStream
      .outputMode("complete")
      .format("console")
      .trigger(Trigger.ProcessingTime("120 seconds"))
      .option("path", "hdfs://dicvmd2.ie.cuhk.edu.hk:8020/user/s1155160788/output/")
      .option("checkpointLocation", "hdfs://dicvmd2.ie.cuhk.edu.hk:8020/user/s1155160788/checkpoint")
      .option("truncate", "false")
      .option("numRows", 30)
      .start()
      .awaitTermination()
  }
}
```

Here, I attach some screenshots of my considerations from the Blackboard forums, as well as the corresponding TA response.



I have spent a good amount of time trying to satisfy the requirements of this assignment in question 3 but have been unable to so far.

It may not be possible with the Structured Streaming API, at least in Spark version 2.3.0 which the IE DIC cluster runs.

I can sum up the requirements as I understand them as such: Output the top 30 most frequent hashtags for every time window. The time window is 5 minutes and the sliding interval is 2 minutes.

I am at the point where I can filter the hashtags, and have preserved the kafka timestamps to enable windowed operations under structured streaming.

The setup:

- To complete the assignment, we need to aggregate through something like count(), and then sort the results into the top 30 in descending order.
- According to the Spark Structured Streaming Programming guide for Spark 2.3.0 (which the cluster runs), sort is only possible when the output mode for .writeStream is set to complete, so update and append output modes, and any operations involving them are not possible.
- Based on the programming guide, you cannot return single counts on a streaming dataset, and need to use groupBy in combination with count. Every window operation example in the guide either uses groupBy with window at some point.
- If we groupby Window alone, it will count the number of tuples per window, when we are trying to count hashtags. If we group by both windows and hashtags, we get thr columns, the time window, the tag, and the count, which is good, so we need to group by both window and hashtags.

The problem:

- Under the constraints above, we do not get a list of hashtags and their counts per time window / sliding window, ready to sort, unlike the spark streaming RDD. We have a list of hashtags and counts, but also their corresponding window, and hashtags are repeated whenever they occur naturally in more than one time window. With this setup, we cannot partition / divide the resulting tuples for each sliding window.
- It is possible to order by window, but if you order by Window alone, the hashtag counts are not sorted by value in descending order.
- It is possible to order by both window and value, in which case you get the descending order of hashtags for each window, one window at a time, however it is hard to control the output display.
  - The guide states: "Limit and take first N rows are not supported on streaming Datasets.", so we can't limit the results to 30 per window, instead it will show every hashtag from every window, just in an ordered manner. If we are allowed to manually take the first 30 results from each window, this is a viable option.
  - It is possible to limit the number of lines written to 30 using the numRows option, but then what we have is the 30 most frequently occuring hashtags across every window, as well as a list of the windows in which they occur. There will be frequent reptition here for hashtags like "Bitcoin". This will only show the top 30 hashtags among all tweets ingested since the consumer started however, not the top 30 for each sliding window.

The question:

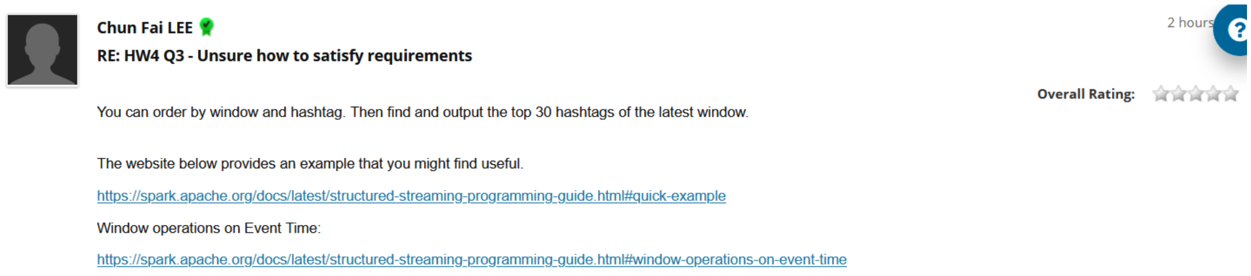
Should we:

1. Order by both window and hashtag and either provide the full output or manually pull out the first 30 for each sliding window.

2. Order by both window and hashtag, limit number of rows written per batch to 30 to provide the top 30 hashtags across the entire consumption period (including repeats from different windows).

3. Something else that I am not sure how to do in order to partition the results by the sliding windows? I searched the internet for several hours but couldn't find anything like this, given the constraints already inherent with the setup and structured streaming under Spark 2.3.0.

Because update and append output modes are not possible without sacrificing sorting + ordering, and limit + take is not possible in general, our options seem limited.



Chun Fai LEE

RE: HW4 Q3 - Unsure how to satisfy requirements

2 hours

Overall Rating: ☆☆☆☆☆

You can order by window and hashtag. Then find and output the top 30 hashtags of the latest window.

The website below provides an example that you might find useful.

https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html#quick-example

Window operations on Event Time:

https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html#window-operations-on-event-time

As you can see, I spent a long time on this part considering all the possible methods. I ultimately group by both the time window and the hashtags, and order by both the time window and hashtags, with the ordering for each being in descending order.

This has the effect of satisfying the TA requirement of showing the latest window, as descending order for window means sorting by where the timestamp is latest, rather than the earliest timestamps. Then, within each window, it shows the hashtags from most frequent occurrence to least.

While in my final code, I specify “.trigger(Trigger.ProcessingTime("120 seconds"))”, which has the effect of making the batch duration 120 seconds, I also experimented with making the batch duration 10 seconds.

I will show output from both approaches below and comment on the pros and cons of each. I am uploading text files for these outputs like I did for Question 2A, and also linking the full stdout logs from the IE DIC cluster. You can view these logs from web interface while connected to GlobalProtect.

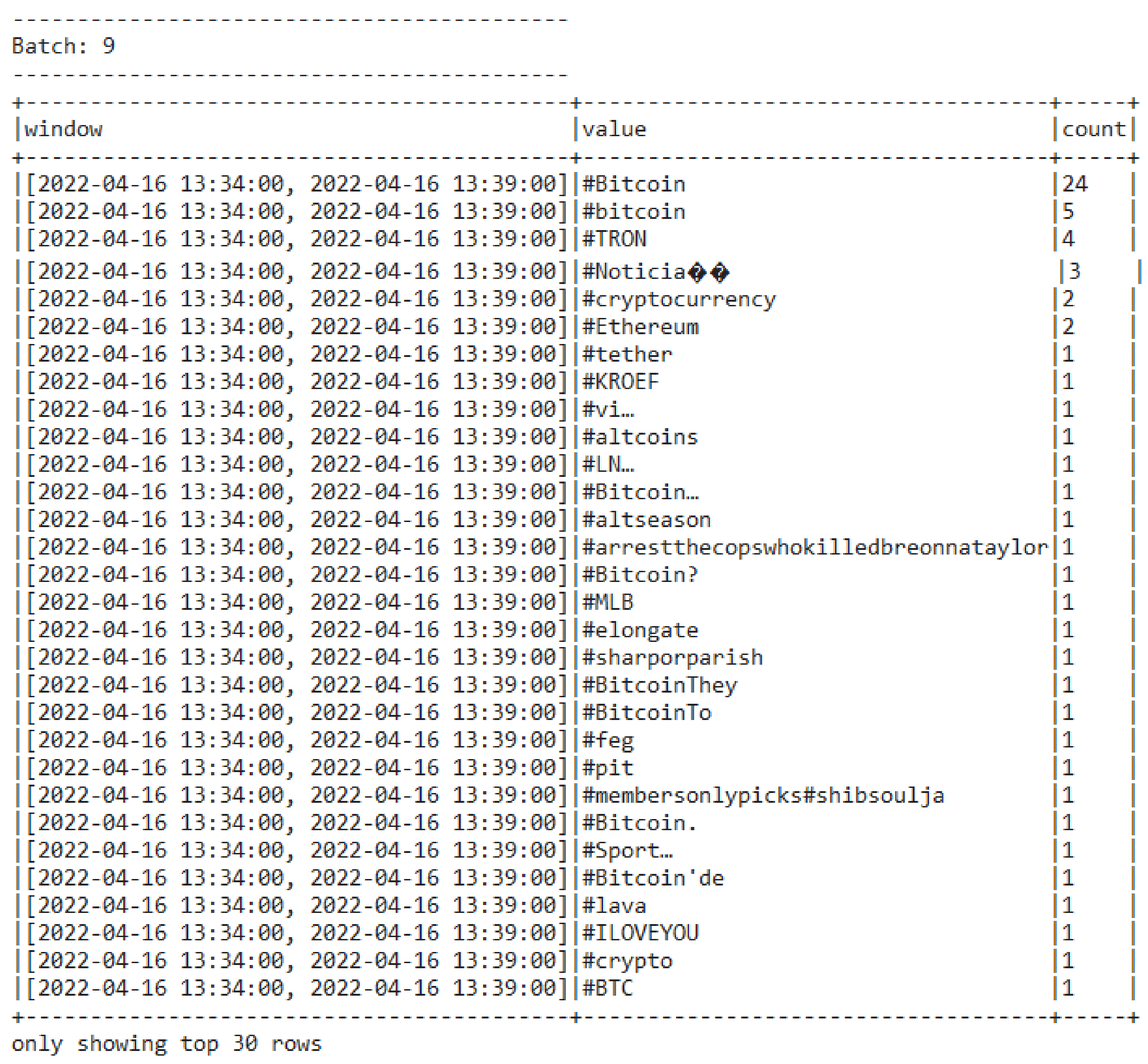
Please see [43] for final output from batch duration of 120 seconds.
http://dicvmd2.ie.cuhk.edu.hk:19888/jobhistory/logs/dicvmd10.ie.cuhk.edu.hk:45454/container_e11_1644821811052_5372_01_000001/container_e11_1644821811052_5372_01_000001/s1155160788/stdout/?start=0

Please see [44] for final output from batch duration of 10 seconds.
http://dicvmd2.ie.cuhk.edu.hk:19888/jobhistory/logs/dicvmd5.ie.cuhk.edu.hk:45454/container_e11_1644821811052_5378_01_000001/container_e11_1644821811052_5378_01_000001/s1155160788/stdout?start=0

I went with the first batch duration in my final code, 120 seconds, because it is easier to understand. During Batch 0 and Batch 1, the system is still ramping up and probably performing the necessary pre-calculations / optimizations before starting effective consumption and processing. We could also consider that each piece of data consumed is going through a streaming pipeline that takes time to process. By Batch 2, we start to get the first output that resembles a window.

Because the batch duration is 120 seconds, which is equal to the slide interval, every 120 seconds, and we also have it set to order by the latest window, every batch essentially shows a new sliding window results as long as during that sliding window, the program processed at least 30 distinct hashtags (as the top 30 results are shown). If not, then it will start to show the most frequent hashtags from the sliding window right before it (such as in Batch 8), since our code doesn’t technically eliminate data from other sliding windows, just put it after the primary one. This is fairly rare, however, and only occurred once during testing. This seems to be the best we can do since we are limited to using structured streaming and in light of the restrictions I mentioned before in my forum post.

Sample output from this approach on the next page, please see text files / linked website for full logs.



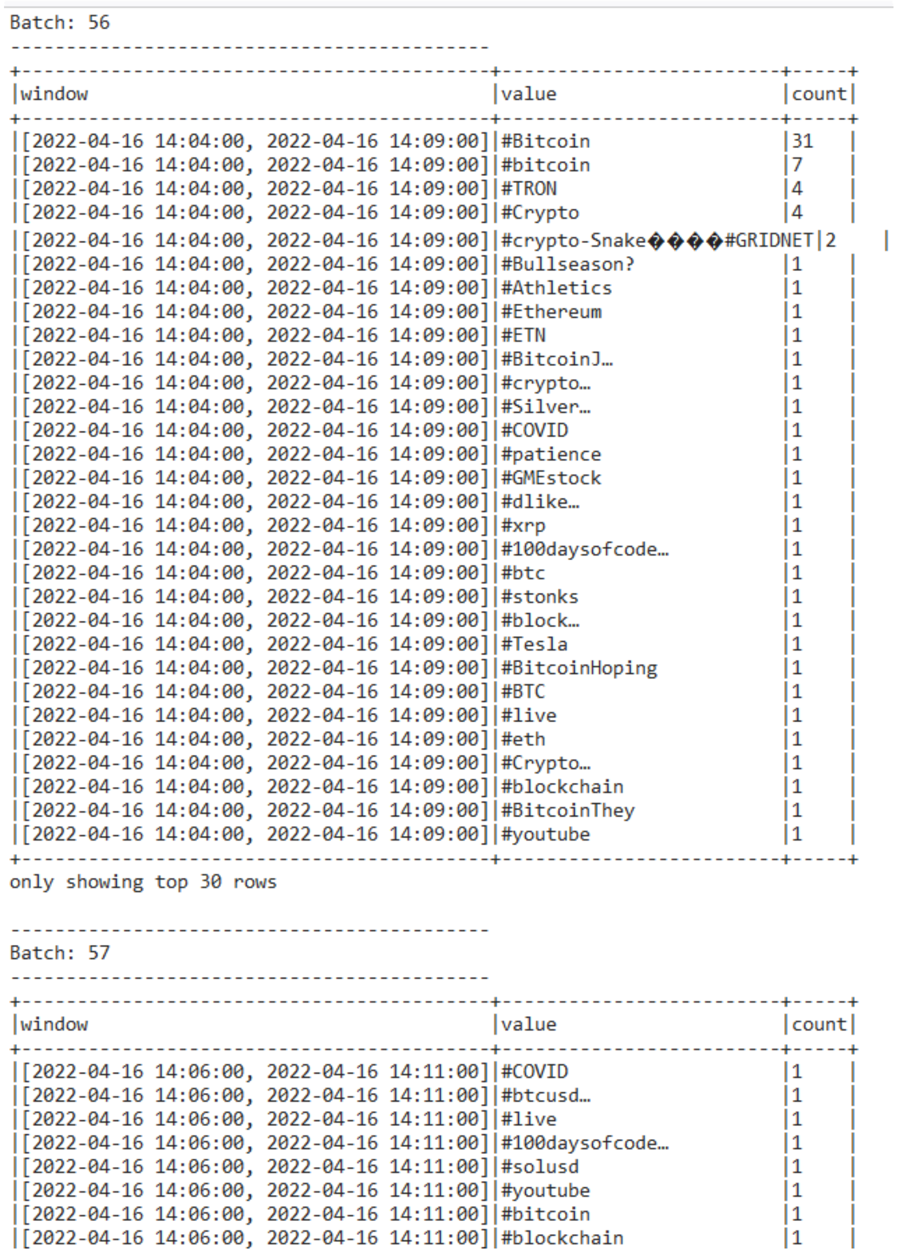
```
-------------------------------------------
Batch: 9
-------------------------------------------
+------------------------------------------+------------------------------------+-----+
|window                                    |value                               |count|
+------------------------------------------+------------------------------------+-----+
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#Bitcoin                            |24   |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#bitcoin                            |5    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#TRON                               |4    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#Noticia��                          |3    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#cryptocurrency                     |2    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#Ethereum                           |2    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#tether                             |1    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#KROEF                              |1    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#vi…                                |1    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#altcoins                           |1    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#LN…                                |1    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#Bitcoin…                           |1    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#altseason                          |1    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#arrestthecopswhokilledbreonnataylor|1    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#Bitcoin?                           |1    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#MLB                                |1    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#elongate                           |1    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#sharporparish                      |1    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#BitcoinThey                        |1    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#BitcoinTo                          |1    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#feg                                |1    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#pit                                |1    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#membersonlypicks#shibsoulja        |1    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#Bitcoin.                           |1    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#Sport…                             |1    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#Bitcoin'de                         |1    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#lava                               |1    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#ILOVEYOU                           |1    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#crypto                             |1    |
|[2022-04-16 13:34:00, 2022-04-16 13:39:00]|#BTC                                |1    |
+------------------------------------------+------------------------------------+-----+
only showing top 30 rows
```

One potential weakness of the 120 second batch duration approach is that we may not be sampling at the very end of a sliding window due to imprecise alignment between the sliding time windows and the batch duration. For example, during the window of batch 9 between 1:34 and 1:39 PM, the tuples being shown are all from the time range, but if the batch duration triggers the processing of this sliding window at 1:38 PM, then the results from 1:38-1:39 PM are not included. This issue is unfortunate, but we don't have a way to accurately output the results of a single time window unlike the Spark Streaming RDD interface so this is a workaround.

Recognizing this limitation, I also experimented with 10 second batch duration, identical to that of question 2. This output is more verbose and difficult to understand, with the benefit of potentially being more accurate. In this output, there will frequently be tuples from multiple windows, because when one sliding window ends and the next has started, there will barely be anything processed on the new window, so most of the top 30 frequent results have to be from the previous window. If the new sliding window has processed only 3 distinct hashtags, the other

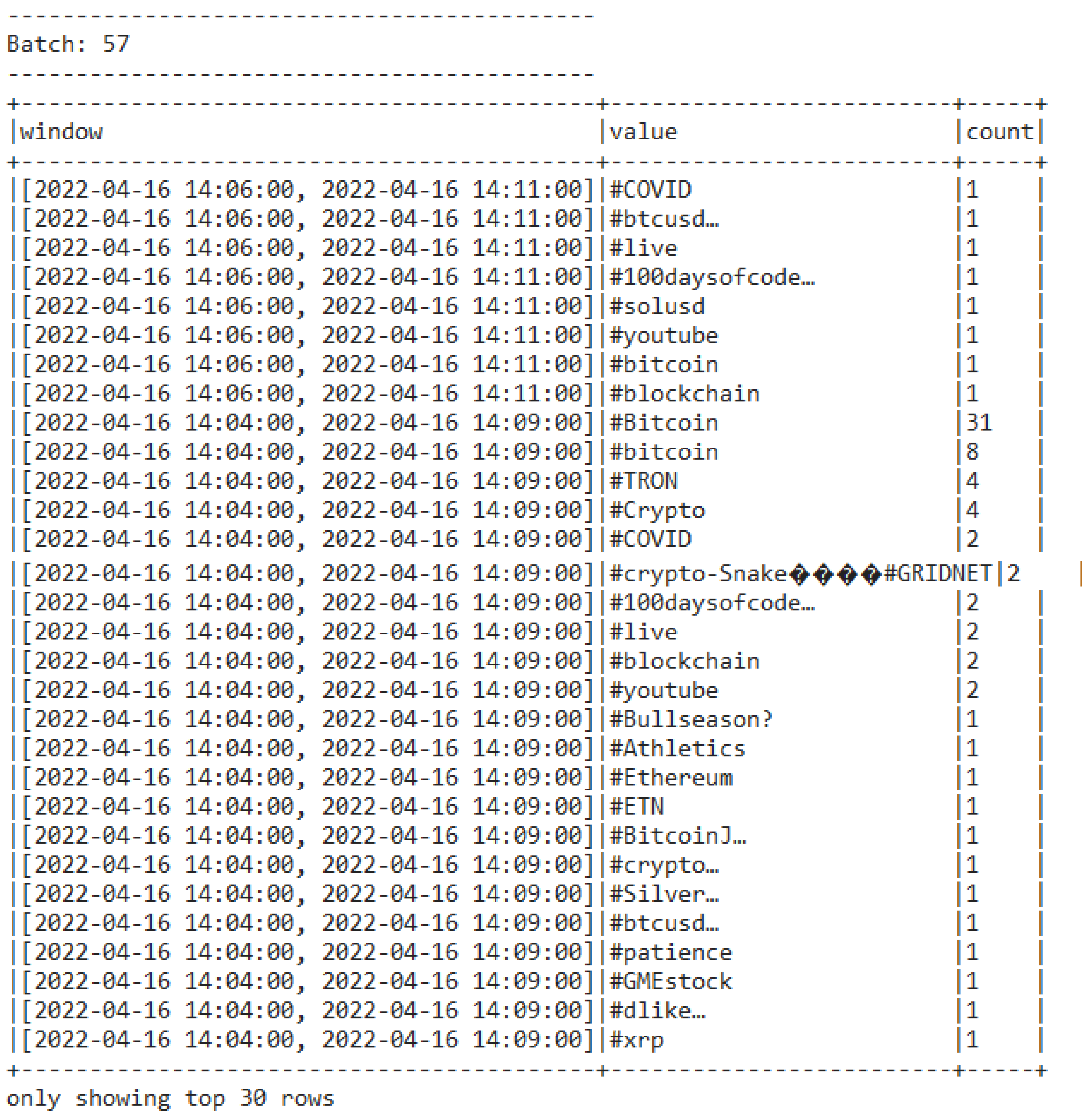
27 are the 27 most frequently occurring hashtags from the previous window. As the batches go on, results from the current sliding window will continue to grow and take a larger and larger share of the total batch output because the ordering explicitly gives preferences to results from the latest window.
To interpret the data, I look at the results from a sliding window "grow" (as in take up more spots) until we start getting results from the next window, then go one batch back, to get the latest results on the sliding window. This approach can only exclude at most 10 seconds of results from any sliding window.

Screenshot of output from this method below, with a full sliding window batch as an example. Again, I chose to go with the batch duration = 120 seconds method because even if it is a little less accurate, it is much more comprehensible.



```
Batch: 56
-------------------------------------------
+------------------------------------------+------------------------+-----+
|window                                    |value                   |count|
+------------------------------------------+------------------------+-----+
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#Bitcoin                |31   |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#bitcoin                |7    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#TRON                   |4    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#Crypto                 |4    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#crypto-Snake����#GRIDNET|2    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#Bullseason?            |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#Athletics              |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#Ethereum               |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#ETN                    |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#BitcoinJ…              |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#crypto…                |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#Silver…                |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#COVID                  |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#patience               |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#GMEstock               |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#dlike…                 |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#xrp                    |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#100daysofcode…         |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#btc                    |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#stonks                 |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#block…                 |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#Tesla                  |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#BitcoinHoping          |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#BTC                    |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#live                   |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#eth                    |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#Crypto…                |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#blockchain             |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#BitcoinThey            |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#youtube                |1    |
+------------------------------------------+------------------------+-----+
only showing top 30 rows

-------------------------------------------
Batch: 57
-------------------------------------------
+------------------------------------------+------------------------+-----+
|window                                    |value                   |count|
+------------------------------------------+------------------------+-----+
|[2022-04-16 14:06:00, 2022-04-16 14:11:00]|#COVID                  |1    |
|[2022-04-16 14:06:00, 2022-04-16 14:11:00]|#btcusd…                |1    |
|[2022-04-16 14:06:00, 2022-04-16 14:11:00]|#live                   |1    |
|[2022-04-16 14:06:00, 2022-04-16 14:11:00]|#100daysofcode…         |1    |
|[2022-04-16 14:06:00, 2022-04-16 14:11:00]|#solusd                 |1    |
|[2022-04-16 14:06:00, 2022-04-16 14:11:00]|#youtube                |1    |
|[2022-04-16 14:06:00, 2022-04-16 14:11:00]|#bitcoin                |1    |
|[2022-04-16 14:06:00, 2022-04-16 14:11:00]|#blockchain             |1    |
```

You can also see from this output, I included not only Batch 56, which was the latest full batch of sliding window 14:04:00 – 14:09:00 but also included batch 57 as a demonstration of how results start to come in from the next sliding window.



```
-------------------------------------------
Batch: 57
-------------------------------------------
+------------------------------------------+-------------------------+-----+
|window                                    |value                    |count|
+------------------------------------------+-------------------------+-----+
|[2022-04-16 14:06:00, 2022-04-16 14:11:00]|#COVID                   |1    |
|[2022-04-16 14:06:00, 2022-04-16 14:11:00]|#btcusd…                 |1    |
|[2022-04-16 14:06:00, 2022-04-16 14:11:00]|#live                    |1    |
|[2022-04-16 14:06:00, 2022-04-16 14:11:00]|#100daysofcode…          |1    |
|[2022-04-16 14:06:00, 2022-04-16 14:11:00]|#solusd                  |1    |
|[2022-04-16 14:06:00, 2022-04-16 14:11:00]|#youtube                 |1    |
|[2022-04-16 14:06:00, 2022-04-16 14:11:00]|#bitcoin                 |1    |
|[2022-04-16 14:06:00, 2022-04-16 14:11:00]|#blockchain              |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#Bitcoin                 |31   |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#bitcoin                 |8    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#TRON                    |4    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#Crypto                  |4    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#COVID                   |2    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#crypto-Snake����#GRIDNET|2    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#100daysofcode…          |2    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#live                    |2    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#blockchain              |2    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#youtube                 |2    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#Bullseason?             |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#Athletics               |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#Ethereum                |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#ETN                     |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#BitcoinJ…               |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#crypto…                 |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#Silver…                 |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#btcusd…                 |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#patience                |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#GMEstock                |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#dlike…                  |1    |
|[2022-04-16 14:04:00, 2022-04-16 14:09:00]|#xrp                     |1    |
+------------------------------------------+-------------------------+-----+
only showing top 30 rows
```

Here I show the full output of batch 57, which is the first batch of a new sliding window. At the top, are the results from the new window, and the rest are populated with the most frequent results from the previous sliding window. You can see the values for some hashtags are higher, such as bitcoin being 8 when it was previously 7. This is because even the batch duration = 10 seconds method can exclude < 10 seconds of results.

References

[1] https://dlcdn.apache.org/kafka/3.1.0/kafka_2.13-3.1.0.tgz

[2] https://kafka.apache.org/quickstart

[3] https://www.learningjournal.guru/article/kafka/installing-multi-node-kafka-cluster/

[4] https://kafka.apache.org/documentation.html#brokerconfigs

[5] https://tmuxcheatsheet.com/

[6] https://www.ibm.com/docs/en/iis/11.7?topic=kafka-setting-up-server

[7] https://stackoverflow.com/questions/60847050/what-is-the-difference-between-advertised-listeners-and-bootstrap-servers

[8] https://mobitec.ie.cuhk.edu.hk/ierg4330/tutorials/09_kafka/

[9] https://www.dropbox.com/s/jdck5tip9v4tzfw/new_tweets.txt?dl=0

[10] https://mobitec.ie.cuhk.edu.hk/ierg4330/static_files/assignments/kafka_producer.py

[11] https://stackoverflow.com/questions/12201928/open-gives-filenotfounderror-ioerror-errno-2-no-such-file-or-directory

[12] https://www.w3schools.com/python/python_strings_slicing.asp

[13] https://stackoverflow.com/questions/18269888/convert-datetime-format-into-seconds

[14] https://www.programiz.com/python-programming/datetime/strptime

[15] https://spark.apache.org/docs/latest/streaming-programming-guide.html#checkpointing

[16] https://www.rittmanmead.com/blog/2017/01/getting-started-with-spark-streaming-with-python-and-kafka/

[17] https://spark.apache.org/docs/2.3.4/streaming-kafka-0-8-integration.html

[18] https://discuss.itversity.com/t/what-does-the-u-stand-for-in-the-rdds-print-thanks/9983/4

[19] https://www.toptal.com/apache/apache-spark-streaming-twitter

[20] https://spark.apache.org/docs/2.3.1/api/python/pyspark.streaming.html

[21] https://spark.apache.org/docs/2.3.0/structured-streaming-kafka-integration.html

[22] https://spark.apache.org/docs/2.3.0/structured-streaming-programming-guide.html

[23] https://spark.apache.org/docs/2.3.0/streaming-programming-guide.html#dataframe-and-sql-operations

[24] https://stackoverflow.com/questions/50768664/what-dependencies-for-structured-streaming-with-kafka

[25] https://stackoverflow.com/questions/50975987/spark-2-3-0-failed-to-find-data-source-kafka

[26] https://medium.com/expedia-group-tech/apache-spark-structured-streaming-output-sinks-3-of-6-ed3247545fbc

[27] https://stackoverflow.com/questions/44583466/exception-in-reading-from-kafka-in-spark-2-1-1-when-using-selectexpr

[28] https://stackoverflow.com/questions/49401572/error-connecting-to-zookeeper-with-spark-streaming-structured

[29] https://stackoverflow.com/questions/52277716/spark-structured-streaming-writestream-to-output-one-global-csv

[30] https://stackoverflow.com/questions/64922560/pyspark-and-kafka-set-are-gone-some-data-may-have-been-missed

[31] https://sparkbyexamples.com/spark/spark-filter-startswith-endswith-examples/

[32] https://stackoverflow.com/questions/55256566/pyspark-selecting-rows-where-column-content-length-x

[33] https://issues.apache.org/jira/browse/SPARK-26379

[34] https://stackoverflow.com/questions/54298251/how-to-include-kafka-timestamp-value-as-columns-in-spark-structured-streaming

[35] http://blog.madhukaraphatak.com/introduction-to-spark-structured-streaming-part-10/

[36] https://stackoverflow.com/questions/36784735/how-to-flatmap-a-nested-dataframe-in-spark

[37] https://sparkbyexamples.com/spark/spark-sql-unix-timestamp/

[38] https://sparkbyexamples.com/spark/spark-dataframe-withcolumn/

[39] https://sparkbyexamples.com/spark/spark-convert-string-to-timestamp-format/

[40] https://databricks.com/blog/2017/05/08/event-time-aggregation-watermarking-apache-sparks-structured-streaming.html

[41] https://smartcat.io/blog/data-engineering/how-to-do-structured-streaming-like-a-boss-part-1/

[42] https://github.com/apache/spark/blob/v3.2.1/examples/src/main/scala/org/apache/spark/examples/sql/streaming/StructuredNetworkWordCountWindowed.scala

[43] http://dicvmd2.ie.cuhk.edu.hk:19888/jobhistory/logs/dicvmd10.ie.cuhk.edu.hk:45454/container_e11_1644821811052_5372_01_000001/container_e11_1644821811052_5372_01_000001/s1155160788/stdout/?start=0

[44] http://dicvmd5.ie.cuhk.edu.hk:8042/node/containerlogs/container_e11_1644821811052_5378_01_000001/s1155160788/stdout/?start=0