**Integration of Spark Streaming with Kafka**

**Reference:**

<https://cloudxlab.com/assessment/displayslide/656/integrating-apache-spark-streaming-apache-kafka-hands-on>

**code reference**

<http://wpcertification.blogspot.com/2017/01/spark-streaming-kafka-10-api-word-count.html>

**Requirements:**

* spark
* git
* jdk 8
* scala
* sbt
* curl
* bigdata folder

**If u need a greater version of ubuntu:**

<https://ubuntu.com/download/desktop/thank-you?version=20.04.3&architecture=amd64>

**Install jdk 8 if not installed**

**sudo apt-get update**

**sudo apt-get install openjdk-8-jre**

**in .bashrc file paste this by your user name**

export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64/

set PATH="$PATH:JAVA\_HOME/bin"

export HADOOP\_HOME=/home/usr/hadoop-2.7.0

export PATH=$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin:$PATH

**#spark**

export SPARK\_HOME=/home/kareem/spark-2.4.0-bin-hadoop2.7

export PATH=$PATH:$SPARK\_HOME/bin

**Step 1:**

**Download scala:**

**SCALA 2.11**

**wget https://downloads.lightbend.com/scala/2.11.12/scala-2.11.12.deb**

**dpkg -i scala-2.11.12.deb**

**scala --version**

**Step 2:**

Install git

sudo apt-get update

sudo apt-get install git

git --version

git clone <https://github.com/cloudxlab/bigdata.git>

Now the repository from git should be in the home directory named as” bigdata”

**Step 3:**

**install scala first without it sbt wont work**

**sudo apt-get install curl**

**SBT**

<https://www.scala-sbt.org/1.x/docs/Installing-sbt-on-Linux.html>

**Install SBT way 1**

**Now it's time to install sbt. First add the necessary repository with the command:**

**echo "deb https://dl.bintray.com/sbt/debian /" | sudo tee -a /etc/apt/sources.list.d/sbt.list**

**Add the public key for the**

**installation with the command:**

**sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 2EE0EA64E40A89B84B2DF73499E82A75642AC823**

**Update apt with the command:**

**sudo apt-get update**

**Finally, install sbt with the command:**

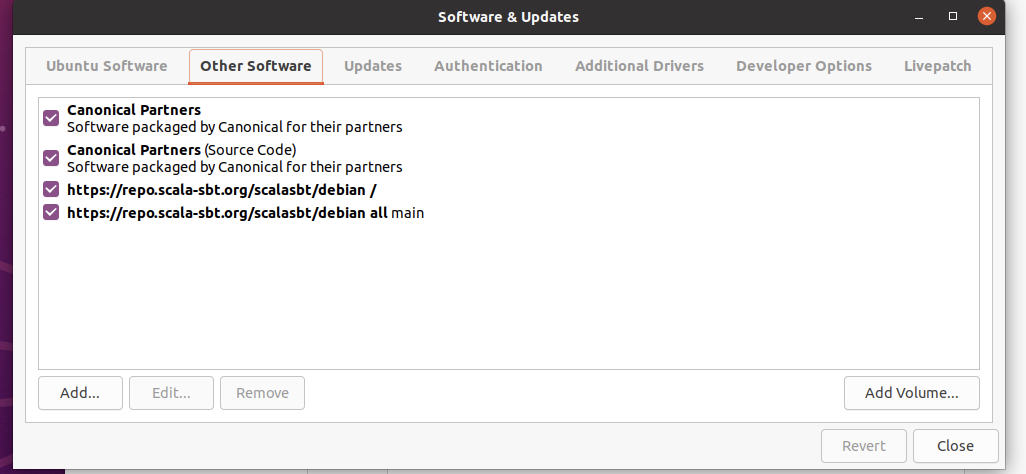
**sudo apt-get install sbt -y**

**Once the installation is complete, test to make sure all is working with the command:**

**sbt test**

**if “unable to locate error” occurs try this**

go to ubuntu settings/about/software updates/



enable all the options shown above.

**if still sbt wont install then try this:**

**sudo apt-get remove sbt**

**Another way to install sbt:**

**echo "deb https://repo.scala-sbt.org/scalasbt/debian all main" | sudo tee /etc/apt/sources.list.d/sbt.list**

**echo "deb https://repo.scala-sbt.org/scalasbt/debian /" | sudo tee /etc/apt/sources.list.d/sbt\_old.list**

**curl -sL "https://keyserver.ubuntu.com/pks/lookup?op=get&search=0x2EE0EA64E40A89B84B2DF73499E82A75642AC823" | sudo apt-key add**

**sudo apt-get update**

**sudo apt-get install sbt**

**sbt test**

**sbt --version /sbt -version**

**select all the options in this page. then check again that error**

**\*note scala should be installed before sbt.**

After installing sbt go to kafka folder in terminal and paste this cmd

**Step 4:**

**run this export PATH…. on terminal\***

export PATH=$PATH:/usr/hdp/current/kafka-broker/bin

**Step 4.1:**

In the file manager go to bigdata folder

**spark/examples/streaming/kafka\_word\_count**

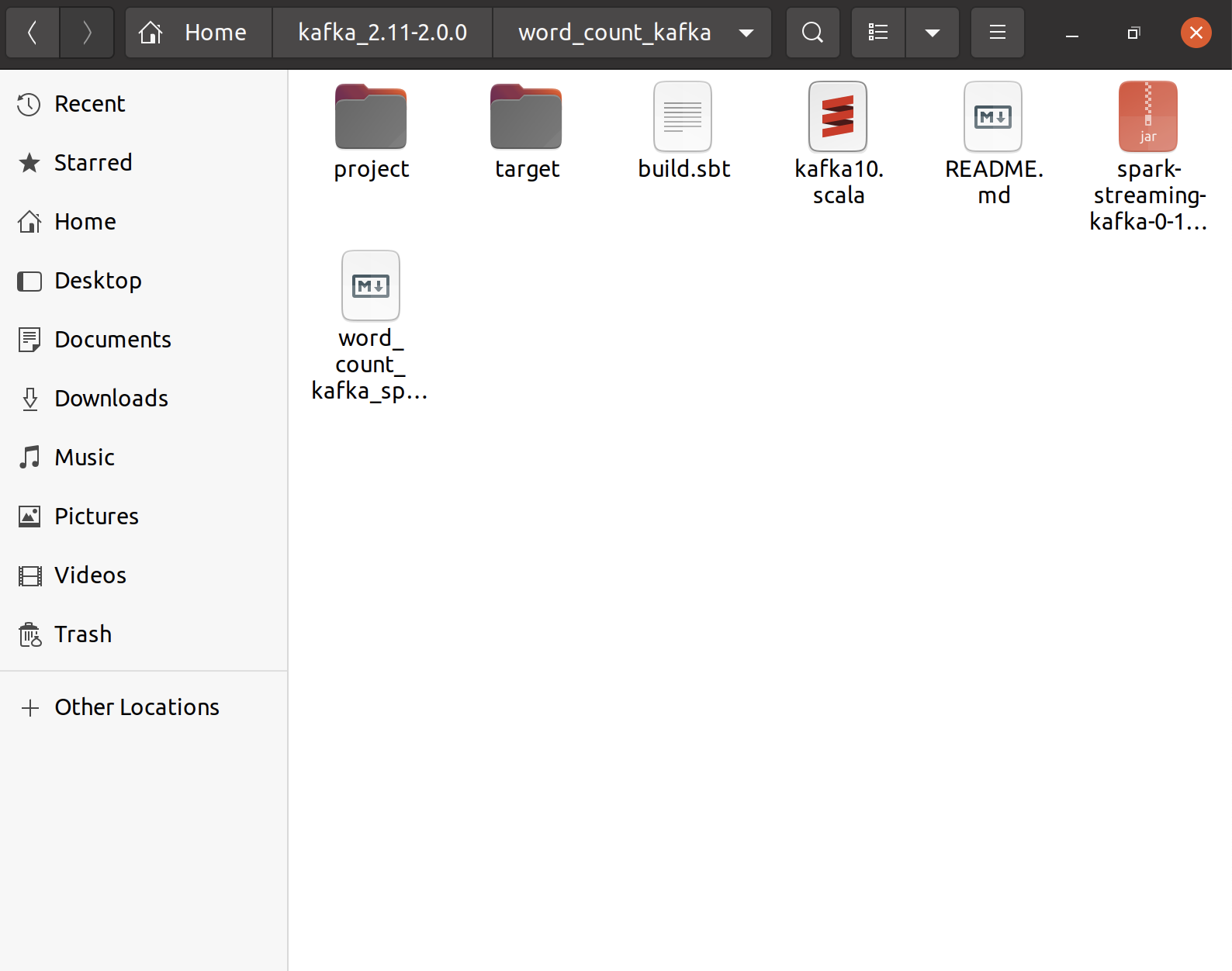
**there will be kafka\_word\_count folder copy it and move it to your kafka folder**

**in this folder you need to replace the code**

**and change the topic name in that “kafka10.scala” code**

**step 4.2:**

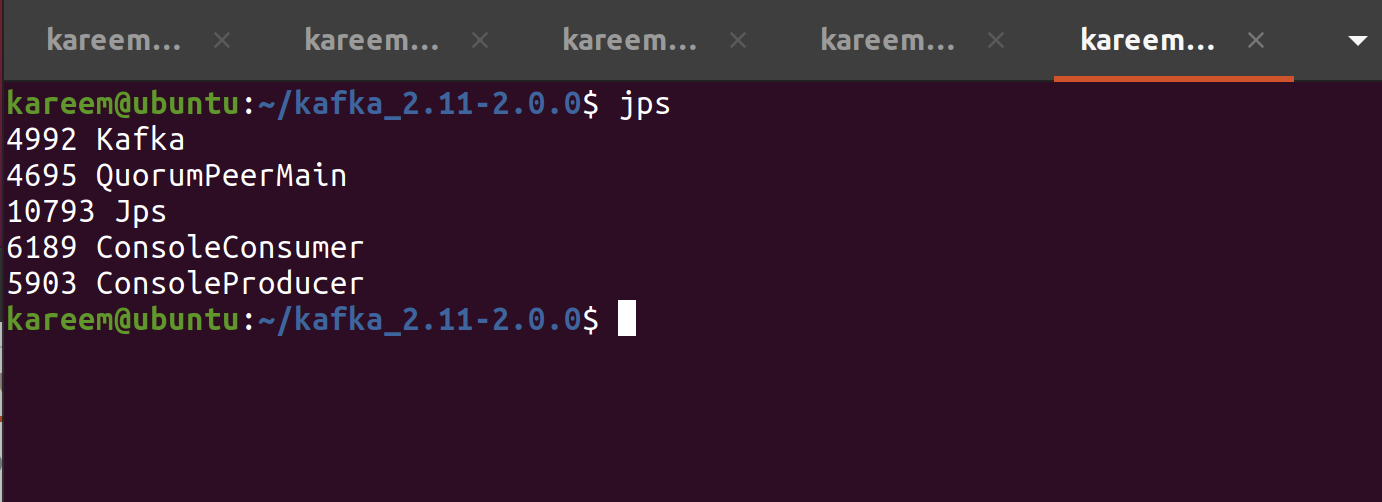
**the folder should like this**

****

**Step 5**

Run  **jps**

*Check kafka and zookeeper(QuorumPeerMain),producer and consumer  is running or not*



*Run kafka zookeeper and create a topic then producer and consumer( normal process of launching kafka).*

*bin/zookeeper-server-start.sh config/zookeeper.properties*

*In new ssh:*

*bin/kafka-server-start.sh config/server.properties*

*In new ssh:*

*bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic kareem*

*bin/kafka-topics.sh --list --zookeeper localhost:2181*

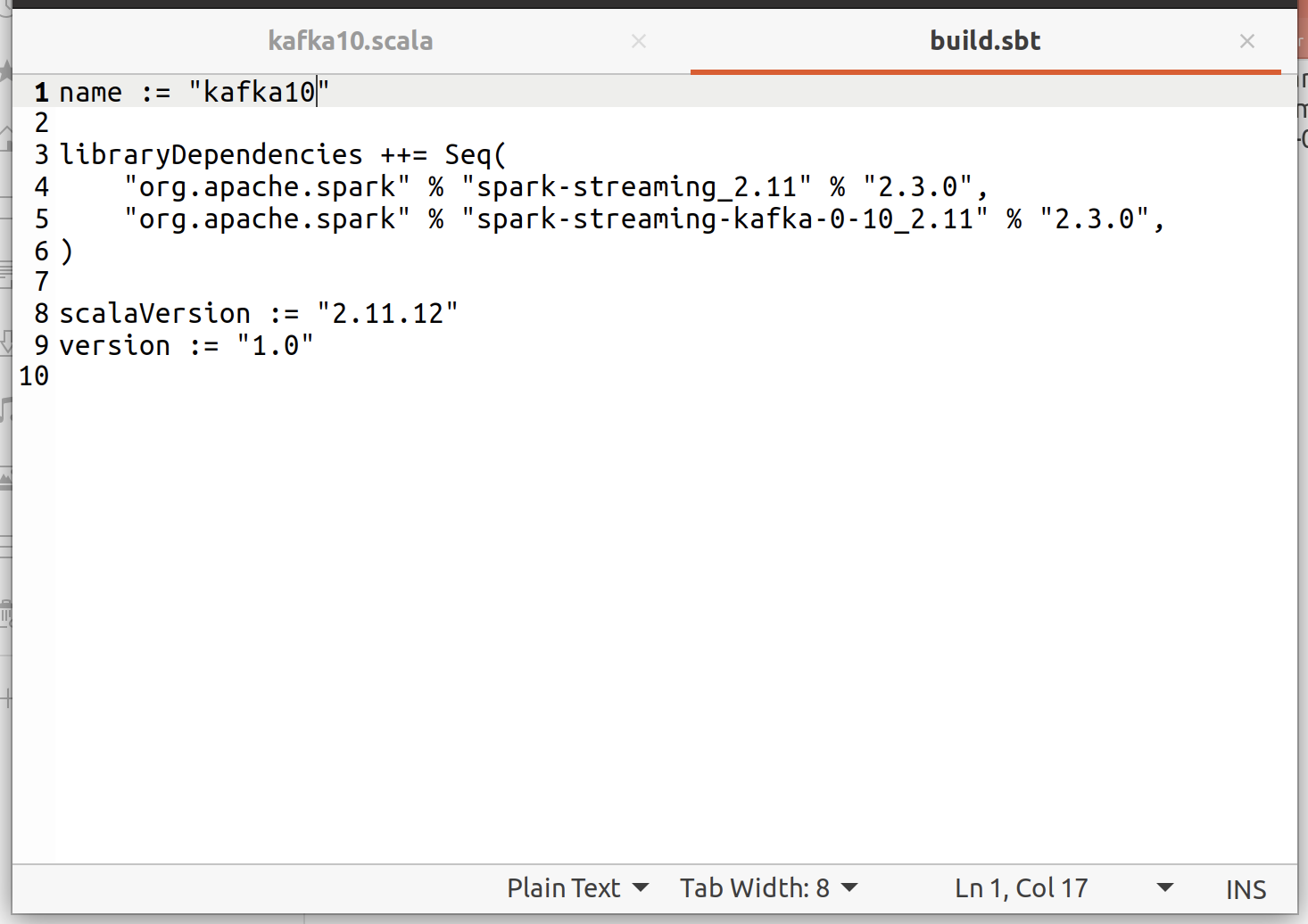
*bin/kafka-console-producer.sh --broker-list localhost:9092 --topic kareem*

*New ssh*

*bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic "kareem" --from-beginning*

**In new terminal go to kafka folder edit the build.sbt inside as name := "kafka10"**

**check the scalaVersion is same as the version in this file**

****

**Next rename the .scala file as kafka10.scala**

**Replace the code in kafka10.scala by this code:**

package com.spnotes.spark

import org.apache.kafka.clients.consumer.ConsumerConfig

import org.apache.spark.SparkConf

import org.apache.spark.streaming.kafka010.{ConsumerStrategies, KafkaUtils, LocationStrategies}

import org.apache.spark.streaming.{Durations, StreamingContext}

import scala.collection.mutable

object Kafka10 {

  def main(argv: Array[String]): Unit = {

    // Configure Spark to connect to Kafka running on local machine

    val kafkaParam = new mutable.HashMap[String, String]()

    kafkaParam.put(ConsumerConfig.BOOTSTRAP\_SERVERS\_CONFIG, "localhost:9092")

    kafkaParam.put(ConsumerConfig.KEY\_DESERIALIZER\_CLASS\_CONFIG,

      "org.apache.kafka.common.serialization.StringDeserializer")

    kafkaParam.put(ConsumerConfig.VALUE\_DESERIALIZER\_CLASS\_CONFIG,

      "org.apache.kafka.common.serialization.StringDeserializer")

    kafkaParam.put(ConsumerConfig.GROUP\_ID\_CONFIG, "group1")

    kafkaParam.put(ConsumerConfig.AUTO\_OFFSET\_RESET\_CONFIG, "latest")

    kafkaParam.put(ConsumerConfig.ENABLE\_AUTO\_COMMIT\_CONFIG, "true")

    val conf = new SparkConf().setMaster("local[2]").setAppName("Kafka10")

    //Read messages in batch of 30 seconds

    val sparkStreamingContext = new StreamingContext(conf, Durations.seconds(30))

    //Configure Spark to listen messages in topic test

    val topicList = List("stream")

    // Read value of each message from Kafka and return it

    val messageStream = KafkaUtils.createDirectStream(sparkStreamingContext,

        LocationStrategies.PreferConsistent,

        ConsumerStrategies.Subscribe[String, String](topicList, kafkaParam))

    val lines = messageStream.map(consumerRecord => consumerRecord.value().asInstanceOf[String])

    // Break every message into words and return list of words

    val words = lines.flatMap(\_.split(" "))

    // Take every word and return Tuple with (word,1)

    val wordMap = words.map(word => (word, 1))

    // Count occurance of each word

    val wordCount = wordMap.reduceByKey((first, second) => first + second)

    //Print the word count

    wordCount.print()

    sparkStreamingContext.start()

    sparkStreamingContext.awaitTermination()

  }

}

In the above code replace the topic name **val topicList = List("stream”)**

to

**val topicList = List(“your\_topic name which u created in topic creation”)**

and save code

**In the kafka\_word\_count folder**

**Open new terminal go to kafka folder**

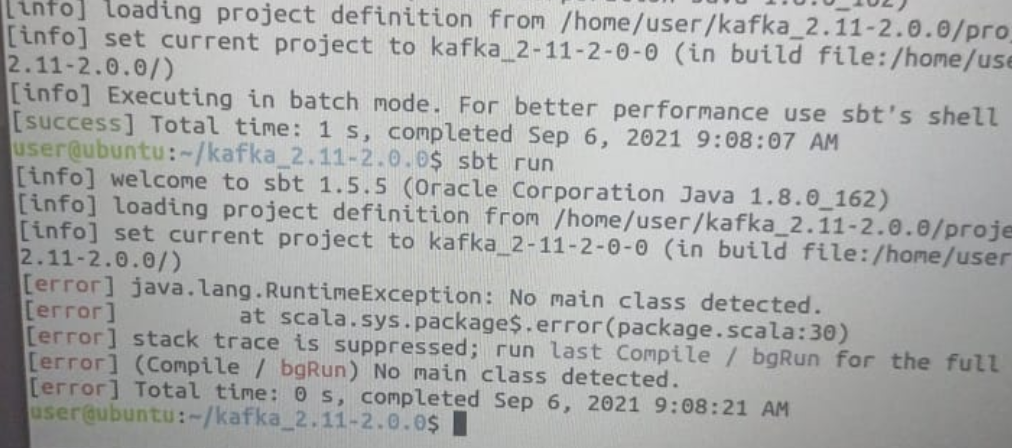
Enter

**go to word\_count\_kafka in terminal then run this**

sbt package

sbt compile    **#this will compile the code wch we previously replaced**

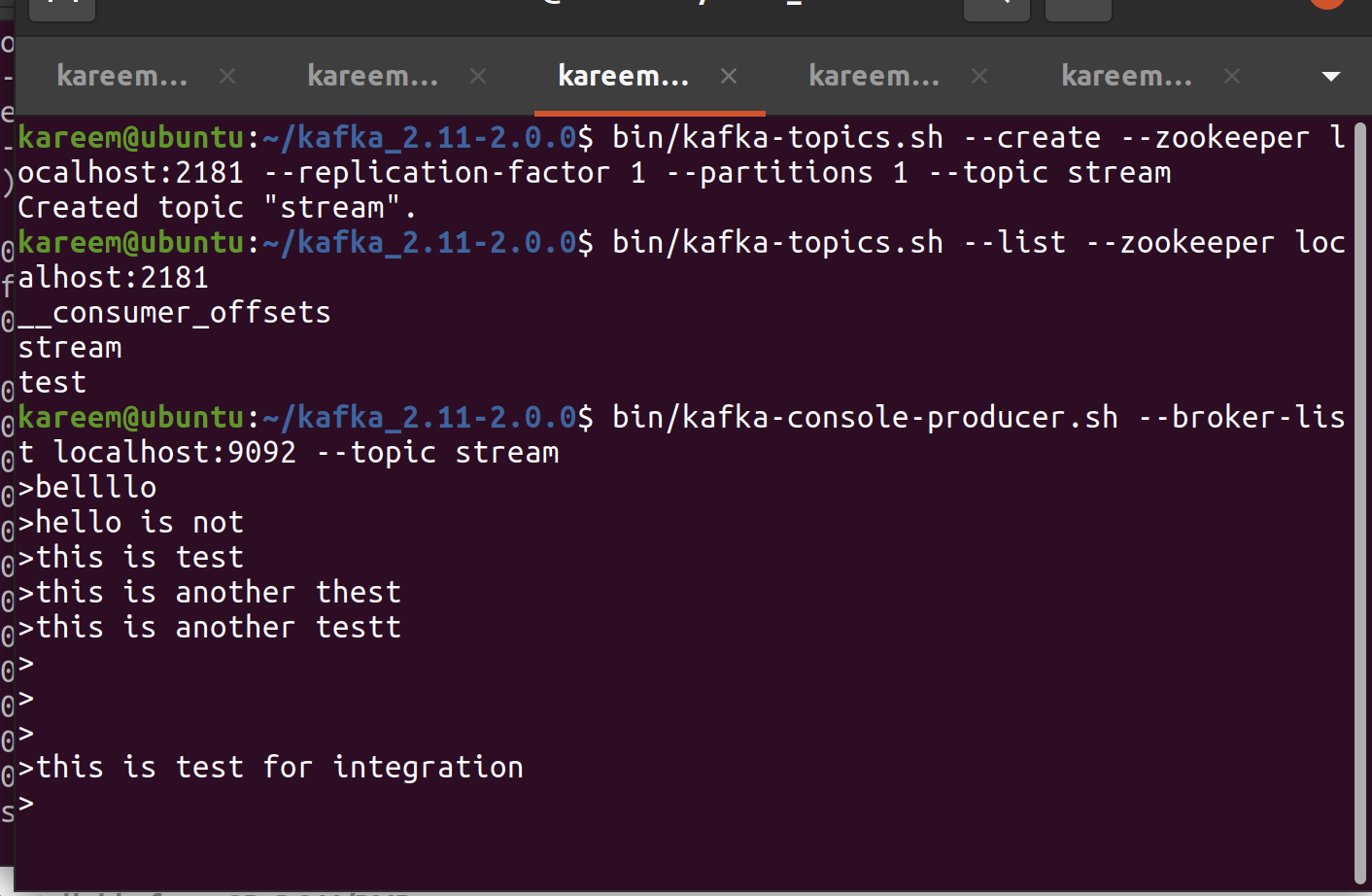
sbt run      **#this will resume the process**

****

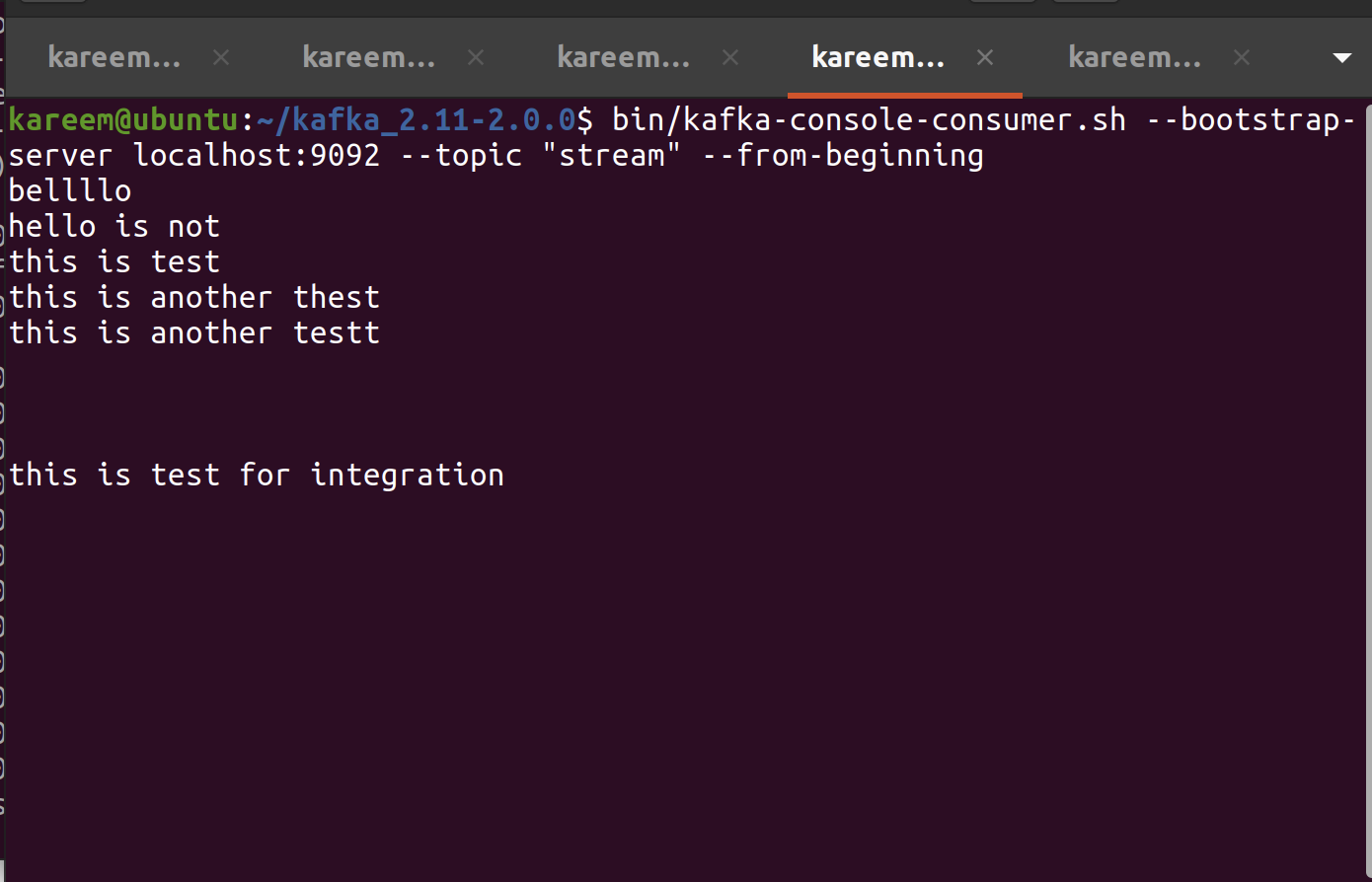
**this issue is for ubuntu version lower than 20.0**

**If “sbt run” won't work check build.sbt**

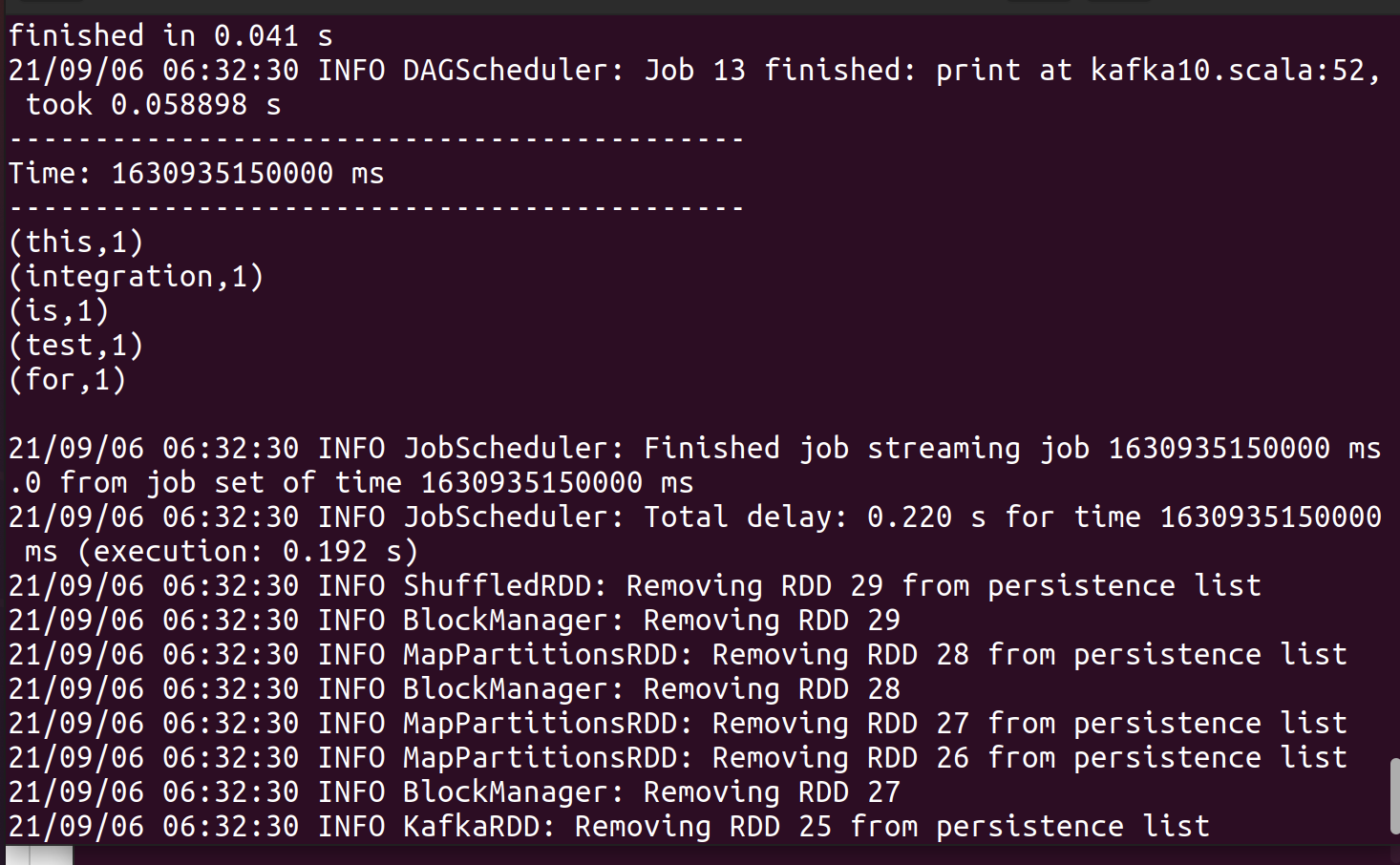
Now then it should show the word count! Of the msgs which are sent to consumer.



**producer**



**consumer**



**output of ”sbt run”.**