Following steps we have to follow

1. First install kafka
2. Set up kafka
3. Start zookeeper
4. Start kafka broker
5. Most important – create kafka topic
6. Then list out some commands to check all kafka topics and all
7. Write producer code and publish messages in kafka topic which we created
8. Write kafka consumer code , where you will pull records from the kafka topic .

Step 1:

Got to this site and identify which version of kafka you are looking

<https://kafka.apache.org/downloads>

run following command to download it

wget <http://apache.claz.org/kafka/2.1.0/kafka_2.11-2.1.0.tgz>

Once download done

Step 2:

Set Kafka home location to PATH environment variable based on your window or MAC it will different

Window I think - .bashrc or .profile

Mac .bash\_profile

vi ~/.bash\_profile

PATH=$PATH:~/kafka/bin

Step 3:

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

Step 4:

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

Step 5:

bin/kafka-topics.sh --create --zookeeper localhost:2181 \

--replication-factor 1 \

--partitions 1 \

--topic News\_reader\_topic

Step 6:

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

Step 7:

Sample scala code

import java.util.Properties

import org.apache.kafka.clients.producer.{KafkaProducer, ProducerRecord}

object KafkaProducerApp extends App {

val props:Properties = new Properties()

props.put("bootstrap.servers","localhost:9092")

props.put("key.serializer","org.apache.kafka.common.serialization.StringSerializer")

props.put("value.serializer","org.apache.kafka.common.serialization.StringSerializer")

props.put("acks","all")

val producer = new KafkaProducer[String, String](props)

val topic = "text\_topic"

try {

for (i <- 0 to 15) {

val record = new ProducerRecord[String, String](topic, i.toString, "My Site is sparkbyexamples.com " + i)

val metadata = producer.send(record)

printf(s"sent record(key=%s value=%s) " +

"meta(partition=%d, offset=%d)\n",

record.key(), record.value(),

metadata.get().partition(),

metadata.get().offset())

}

}catch{

case e:Exception => e.printStackTrace()

}finally {

producer.close()

}

}

For python you can follow this

<https://towardsdatascience.com/kafka-python-explained-in-10-lines-of-code-800e3e07dad1>

Step 8:

Sample kafka consumer code

val kafkaMessageReadDf = sparkSession

.readStream

.format("kafka")

.option("subscribe", "eNews\_reader\_topic")

.option("failOnDataLoss", "false")

.option("startingOffsets", "earliest") // From starting

.load()

.selectExpr(

"CAST(key AS STRING)",

"CAST(value AS STRING)",

"CAST(partition AS INTEGER)",

"CAST(offset AS LONG)",

"CAST(timestamp AS STRING)",

"CAST(timestampType AS INTEGER)",

"CAST(topic AS STRING)"

)

// writing data to my machine locally

val kafkaMessageWriteDf = kafkaMessageReadDf.writeStream

.outputMode("append")

.format("json")

// .trigger(Trigger.ProcessingTime("30 minutes"))

// .trigger(Trigger.Continuous("10 second"))

.trigger(Trigger.Once())

.option("path","file://Learning/scala/data\_out/stremout")

.option("checkpointLocation","file:///data\_out/checkout")

.start()

kafkaMessageWriteDf.awaitTermination(100000)

python

<https://towardsdatascience.com/kafka-python-explained-in-10-lines-of-code-800e3e07dad1>