spark-shell --packages org.apache.spark:spark-streaming-kinesis-asl\_2.12:3.2.0

import org.apache.spark.storage.StorageLevel  
import org.apache.spark.streaming.kinesis.KinesisInputDStream  
import org.apache.spark.streaming.{Seconds, StreamingContext}  
import com.amazonaws.services.kinesis.clientlibrary.lib.worker.InitialPositionInStream  
import org.apache.spark.streaming.kinesis.KinesisInitialPositions.\_  
import org.json4s.DefaultFormats  
import org.json4s.\_  
import org.json4s.jackson.JsonMethods.{ parse, compact, render }  
import org.json4s.string2JsonInput  
implicit val formats: org.json4s.DefaultFormats = DefaultFormats  
  
  
val ssc = new StreamingContext(spark.sparkContext, Seconds(60))  
val sc = spark.sparkContext  
  
val stream = KinesisInputDStream.builder.streamingContext(ssc).streamName("myInputStream").endpointUrl("https://kinesis.ap-northeast-1.amazonaws.com").regionName("ap-northeast-1").initialPosition(new TrimHorizon()).checkpointAppName("myapp").checkpointInterval(Seconds(60)).buildWithMessageHandler(record =>("shardId-000000000000", record.getSequenceNumber, new String(record.getData.array)))

stream.foreachRDD(rdd => {val data = rdd.collect()  
if (data.size > 0) {  
data.map(r => r.\_3).foreach(println)}})

ssc.start

ssc.stop()