import com.amazonaws.services.kinesisfirehose.model.\_  
import com.amazonaws.services.kinesisfirehose.AmazonKinesisFirehoseClient  
import com.amazonaws.regions.Regions  
import java.nio.ByteBuffer  
import com.amazonaws.auth.InstanceProfileCredentialsProvider

def putMessage(stream:String, endPoint:String, data:String): String = {  
 val kinesisFirehoseClient=AmazonKinesisFirehoseClient.builder.withRegion(Regions.US\_EAST\_1).withCredentials((new InstanceProfileCredentialsProvider(false))).build()  
val bdata=new Record().withData(ByteBuffer.wrap(data.getBytes))  
 // Create a PutRecordRequest with an Array[Byte] version of the data  
 val putRecordRequest = new PutRecordRequest().withDeliveryStreamName(stream).withRecord(bdata)  
 // Put the record onto the stream and capture the PutRecordResult  
 val putRecordResult = kinesisFirehoseClient.putRecord(putRecordRequest)  
 return putRecordResult.getRecordId()  
 }  
  
  
stream.foreachRDD(rdd => {val data = rdd.collect()  
if (data.size > 0) {  
data.map(r => r.\_3).foreach(x=> putMessage("**mySparkOutput","https://**firehose.us-east-1.amazonaws.com"**,x)**)}})