通过自定义实现KafkaClientSupplier接口实现从一个kafka集群读取数据，再写入到另一个kafka集群中。主要实现如下：

## 1.自定义实现KafkaClientSupplier接口

package org.feiyu.dataprocess.kafka;  
  
import org.apache.kafka.clients.consumer.Consumer;  
import org.apache.kafka.clients.consumer.KafkaConsumer;  
import org.apache.kafka.clients.producer.KafkaProducer;  
import org.apache.kafka.clients.producer.Producer;  
import org.apache.kafka.common.serialization.ByteArrayDeserializer;  
import org.apache.kafka.common.serialization.ByteArraySerializer;  
import org.apache.kafka.streams.KafkaClientSupplier;  
  
import java.util.HashMap;  
import java.util.Map;  
  
/\*\*  
 \* 自定义KafkaClientSupplier  
 \*/  
public class MyKafkaClientSupplier implements KafkaClientSupplier {  
 //生产者配置文件  
 private Map<String,Object> producerConfig;  
 //消费者配置文件  
 private Map<String,Object> consumerConfig;  
  
 public MyKafkaClientSupplier(Map<String, Object> producerConfig, Map<String, Object> consumerConfig) {  
 this.producerConfig = producerConfig;  
 this.consumerConfig = consumerConfig;  
 }  
  
 @Override  
 public Producer<byte[], byte[]> getProducer(Map<String, Object> config) {  
 Map<String,Object> map = new HashMap<>(config);  
 map.putAll(producerConfig);  
 return new KafkaProducer<>(map,new ByteArraySerializer(),new ByteArraySerializer());  
 }  
  
 @Override  
 public Consumer<byte[], byte[]> getConsumer(Map<String, Object> config) {  
 Map<String,Object> map = new HashMap<>(config);  
 map.putAll(consumerConfig);  
 return new KafkaConsumer<>(map,new ByteArrayDeserializer(),new ByteArrayDeserializer());  
 }  
  
 @Override  
 public Consumer<byte[], byte[]> getRestoreConsumer(Map<String, Object> config) {  
 Map<String,Object> map = new HashMap<>(config);  
 map.putAll(consumerConfig);  
 return new KafkaConsumer<>(map,new ByteArrayDeserializer(),new ByteArrayDeserializer());  
 }  
}

## 2.创建自定义的KafkaClientSupplier实例并传入KafkaStreams构造器中

StreamsConfig config = new StreamsConfig(props);  
  
 MyKafkaClientSupplier supplier = new MyKafkaClientSupplier(  
 Collections.singletonMap(ProducerConfig.BOOTSTRAP\_SERVERS\_CONFIG,"//生产者kafka集群配置"),  
 Collections.singletonMap(ConsumerConfig.BOOTSTRAP\_SERVERS\_CONFIG,"//消费者kafka集群配置"));  
  
 KafkaStreams streams = new KafkaStreams(builder, config,supplier);  
 streams.start();  
 kafkaStreams.add(streams);

## 3.在kafka stream启动日志中可以看到配置生效，实现了kafka stream的读写分离

 