1. 生产环境地址：
   kafka版本：kafka_2.13-2.4.0
   broker地址：10.166.139.204:9092,10.166.139.205:9092,10.166.139.206:9092
2. kafka认证及授权
   kafka生产环境已经完成认证及鉴权配置，不允许随意接入kafka环境进行操作。
   使用时需要先由中台管理人员初始化访问帐号及topic，同时对topic进行授权。授权成功后才允许进行集成。
3. 使用kafka-client方式：
   3.1 生产者

```
String mqusername = ""; // 该帐号由中台分配
String password = ""; // 该密码由中台分配
String topicName = ""; // 由中台进行管理同时已经授权
Properties props = new Properties();
props.put("bootstrap.servers",
"10.166.139.204:9092,10.166.139.205:9092,10.166.139.206:9092");
props.put("security.protocol", "SASL_PLAINTEXT"); // 必需，值必须为SASL_PLAINTEXT
props.put("sasl.mechanism", "SCRAM-SHA-256"); // 必需，值必须为SCRAM-SHA-256
props.put("sasl.jaas.config",
 "org.apache.kafka.common.security.scram.ScramLoginModule required
username='"+mqusername+"' password='"+mqpassword+"';");
while (true) {
 ProducerRecord<String, String> record = new ProducerRecord<>(topicName, "topic_" +
System.nanoTime(), System.nanoTime() + " ABCDEFG");
 try {
     producer.send(record).get();
     System.out.println("发送成功! " + i + " " + record);
 } catch (Exception e) {
     e.printStackTrace();
     break;
 }
}
```

3.2 消费者

```
String mqusername = ""; // 该帐号由中台分配
String password = ""; // 该密码由中台分配
String topicName = ""; // 由中台进行管理同时已经授权
Properties props = new Properties();
props.put("bootstrap.servers",
"10.166.139.204:9092,10.166.139.205:9092,10.166.139.206:9092");
props.put("security.protocol", "SASL_PLAINTEXT"); // 必需，值必须为SASL_PLAINTEXT
props.put("sasl.mechanism", "SCRAM-SHA-256"); // 必需，值必须为SCRAM-SHA-256
props.put("sasl.jaas.config",
 "org.apache.kafka.common.security.scram.ScramLoginModule required
username='"+mqusername+"' password='"+mqpassword+"';");
KafkaConsumer<String, String> consumer = new KafkaConsumer(props);
```

```
consumer.subscribe(Arrays.asList(topicName));
while (true) {
 try {
     ConsumerRecords<String, String> records = consumer.poll(100);
     for (ConsumerRecord<String, String> record : records) {
         System.out.printf("partition= %d, offset = %d, key = %s, value = %s\n",
record.partition(),
                 record.offset(), record.key(), record.value());
     }
     consumer.commitSync();
 } catch (Exception e) {
     e.printStackTrace();
     break;
 }
}
```

4 使用kafkaTemplate方式

注意：以下集成方式使用的spring-kafka依赖版本为2.5.11.RELEASE，如果使用其它版本，请检查org.springframework.boot.autoconfigure.kafka.KafkaProperties类中的属性配置与本文档是否一致。

配置文件中的username和password由中台进行管理配置。

4.1 生产者

```
spring:
 kafka:
     bootstrap-servers: ${producer-brokers}
     producer:
         security:
             protocol: SASL_PLAINTEXT
         properties:
             sasl:
                 mechanism: SCRAM-SHA-256
                 jaas:
                     config: org.apache.kafka.common.security.scram.ScramLoginModule
required username='xxx' password='xxx';
```

4.2 消费者

```
spring:
 kafka:
     bootstrap-servers: ${consumer-brokers}
     consumer:
         security:
             protocol: SASL_PLAINTEXT
         properties:
             sasl:
                 mechanism: SCRAM-SHA-256
                 jaas:
                     config: org.apache.kafka.common.security.scram.ScramLoginModule
required username='xxx' password='xxx';
```

5 使用spring-kafka-stream方式
核心配置参考kafkaTemplate方式

6 POM文件依赖

```
<dependency>
    <groupId>org.springframework.kafka</groupId>
    <artifactId>spring-kafka</artifactId>
    <exclusions>
        <exclusion>
            <groupId>org.apache.kafka</groupId>
            <artifactId>kafka-clients</artifactId>
        </exclusion>
        <exclusion>
            <groupId>org.apache.kafka</groupId>
            <artifactId>kafka-streams</artifactId>
        </exclusion>
    </exclusions>
</dependency>
<dependency>
    <groupId>org.apache.kafka</groupId>
    <artifactId>kafka-clients</artifactId>
    <version>2.4.0</version>
</dependency>
<dependency>
    <groupId>org.apache.kafka</groupId>
    <artifactId>kafka_2.13</artifactId>
    <version>2.4.0</version>
</dependency>
```

7python集成方式

```
from kafka import KafkaConsumer
KAFKA_TOPIC = 'test_periodic_data'
KAFKA_BROKERS = '10.166.139.204:9092,10.166.139.205:9092,10.166.139.206:9092'
```

```
consumer = KafkaConsumer(KAFKA_TOPIC,
                         bootstrap_servers=KAFKA_BROKERS,
                         sasl_plain_username="its_ai_mp",
                           sasl_plain_password="Sdhsg2021",
                           security_protocol="SASL_PLAINTEXT",
                           sasl_mechanism="SCRAM-SHA-256")
i = 0
Plate_list = []

for message in consumer:
    print("%s:%d:%d: key=%s value=%s" % (message.topic,
message.partition,message.offset, message.key,message.value))
    print(message)
```