课程实验五：Flink消费Kafka数据

实验时间：2021年05月08日

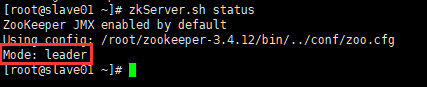
学生姓名：李志毅

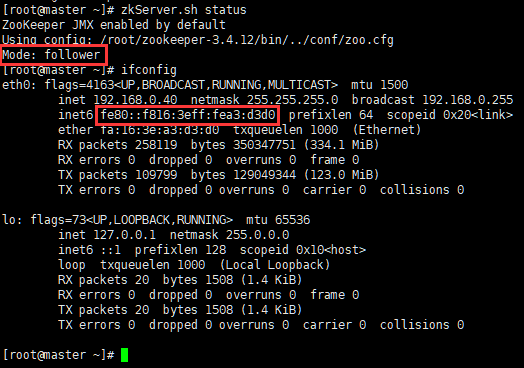
学生班号、学号：2018211314班 2018211582号

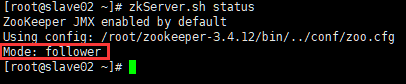
1. **实验结果截图**

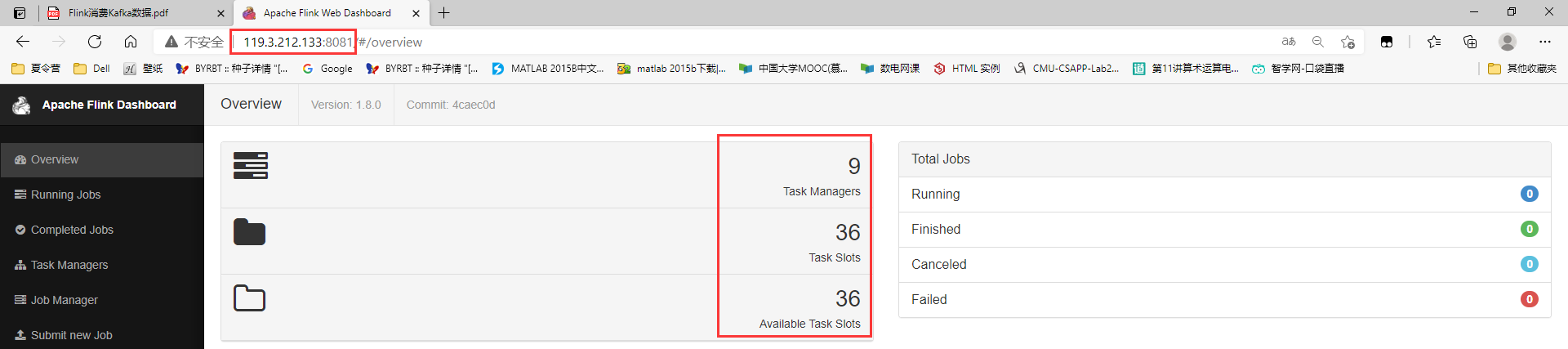
注意：以下每一个结果截图，都必须需要包含所圈中标记信息（通过标记信息，来判别作业是独立完成的）

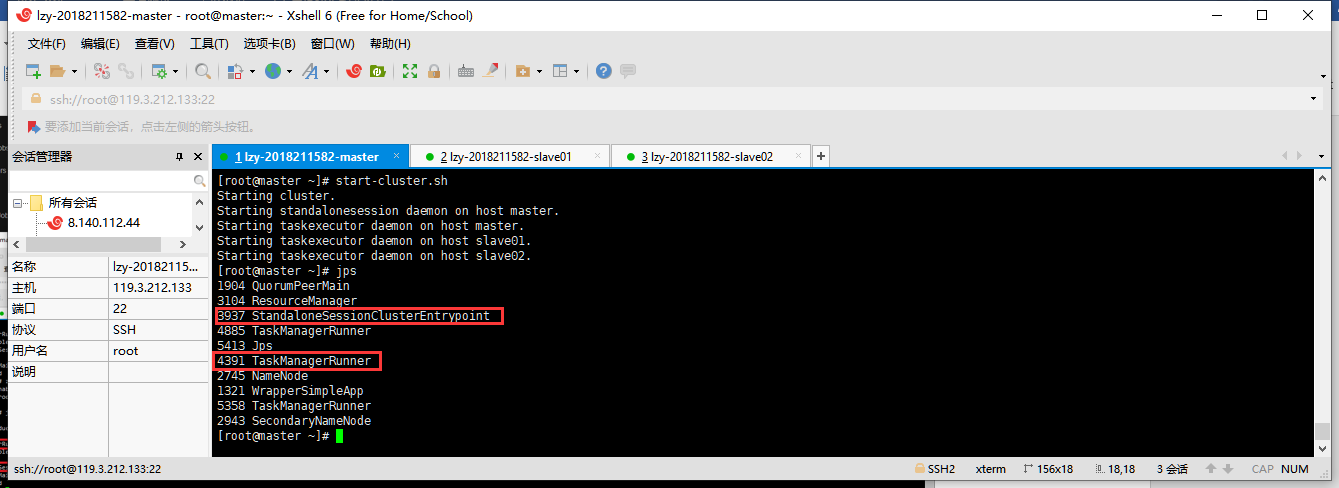
【结果截图1】Zookeeper安装验证



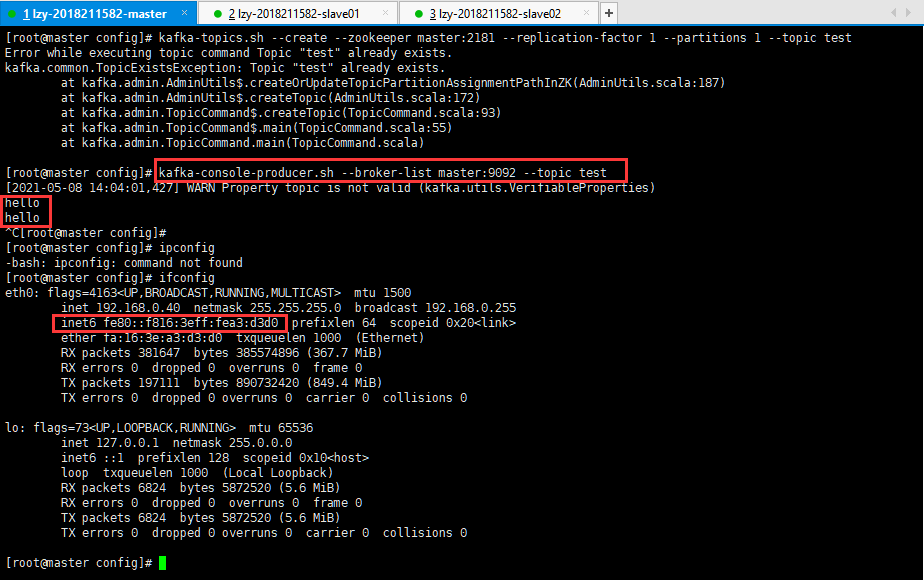


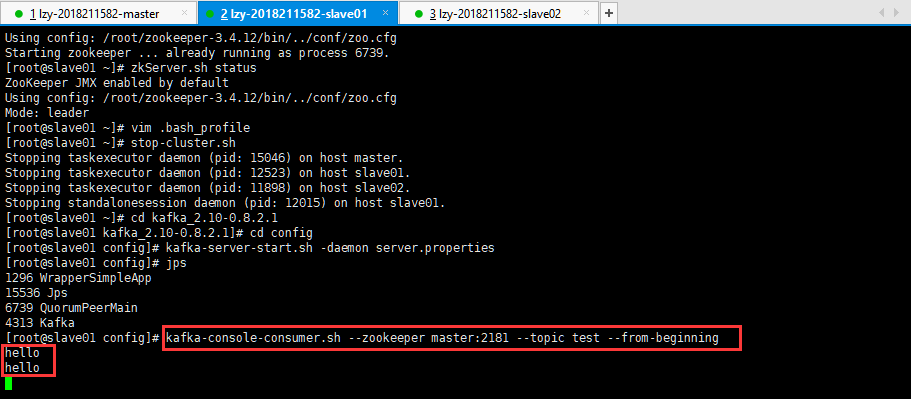


【结果截图2】Flink安装验证

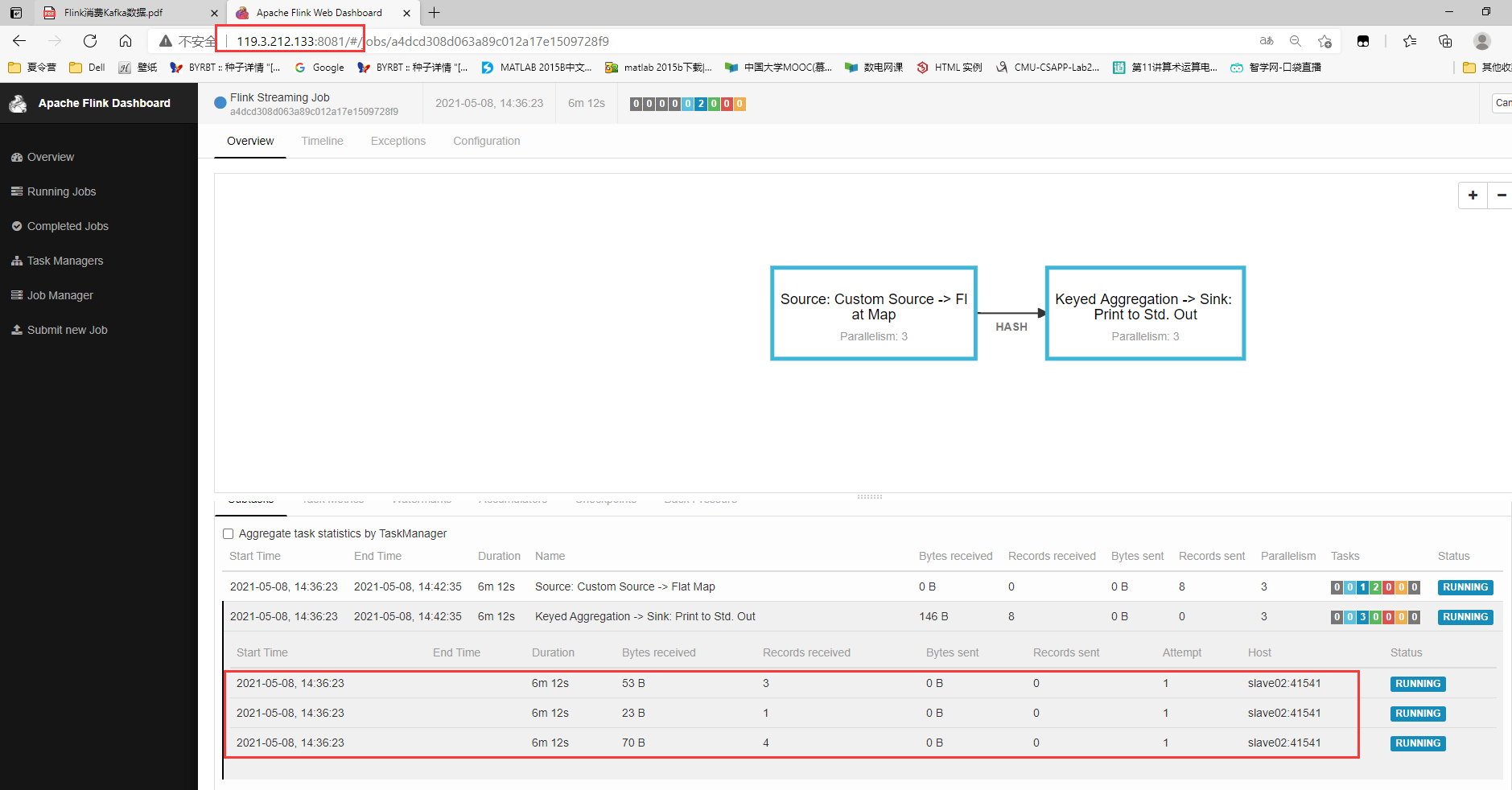


【结果截图3】Kafka对话验证

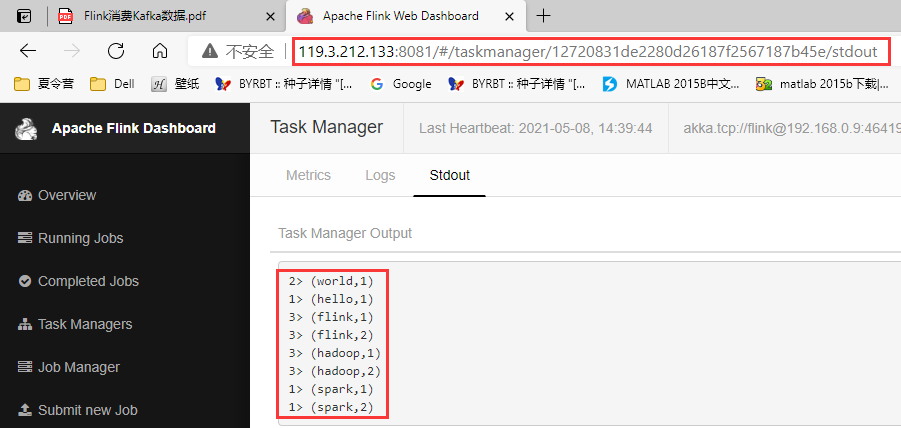


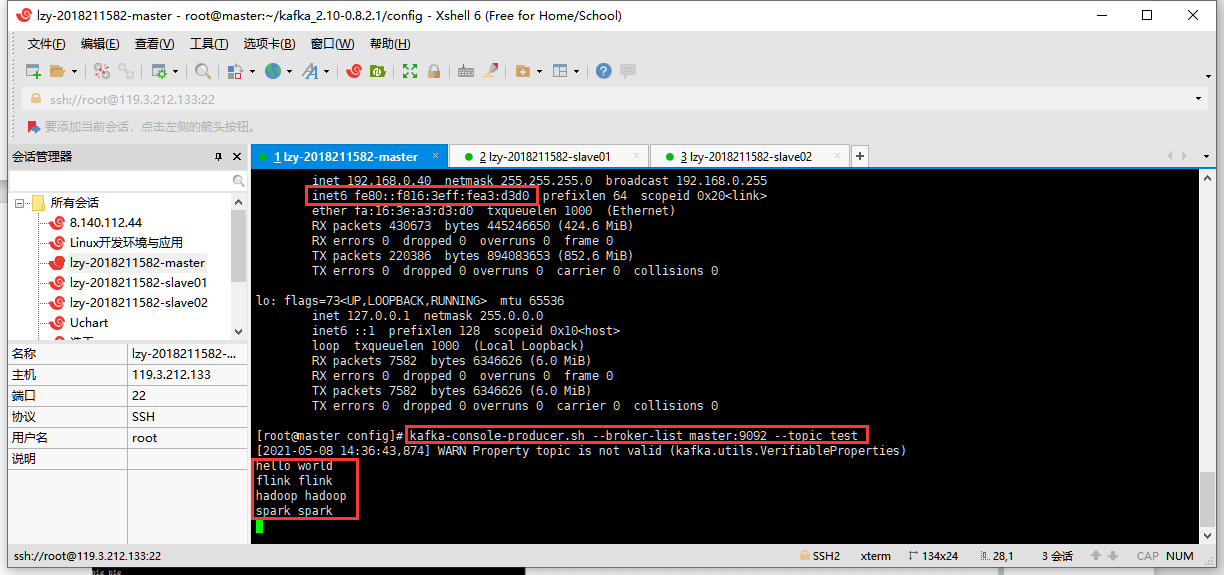


【结果截图4】查询输出结果节点



【结果截图5】单词计数





1. **简要描述实验做了哪些工作？**

实验在之前实验的基础上，经历了以下步骤：

1.安装并部署了Zookeeper，启动了Zookeeper集群

2.安装并启动了Flink

3.安装并配置了Kafka

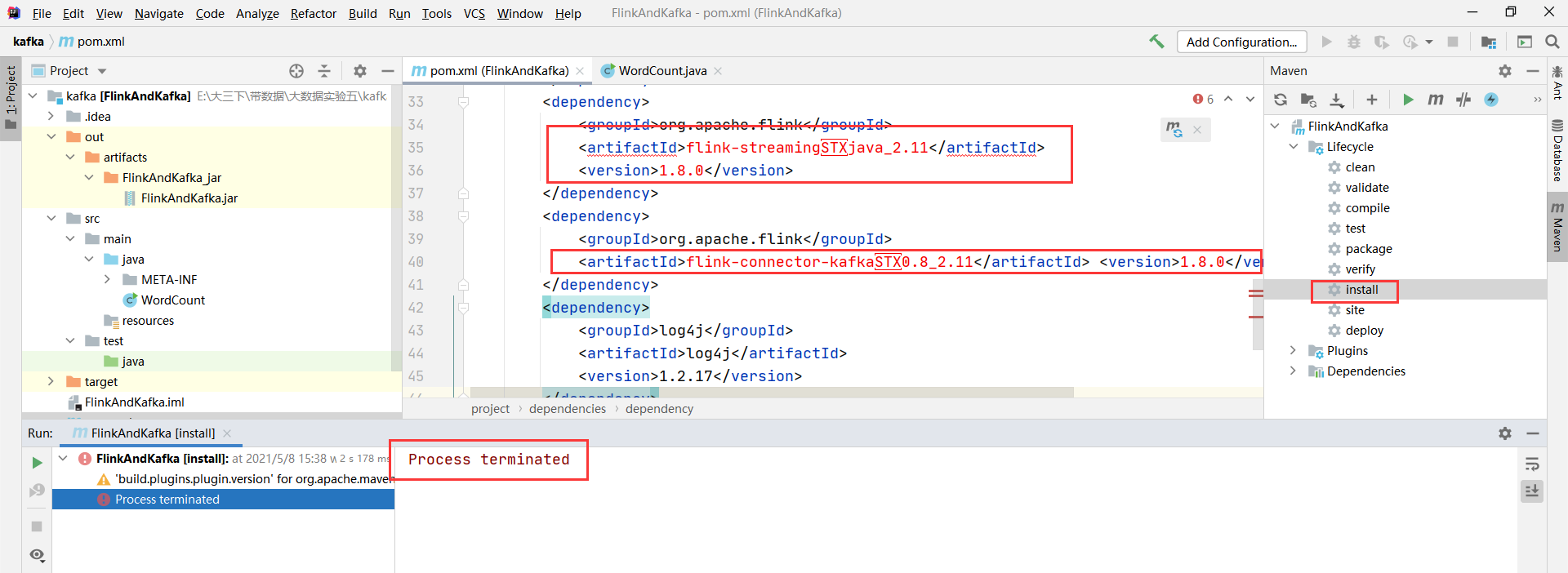
4.创建topic主题，在master启动生产者，在slave01启动消费者，并进行了对话测试

5.编写WordCount程序，使用Flink启动WordCount程序，启动生产者并输入数据，Flink实时从Kafka数据流中获得消息，并进行词频计算

**三.实验过程中遇到的问题和解决办法？**

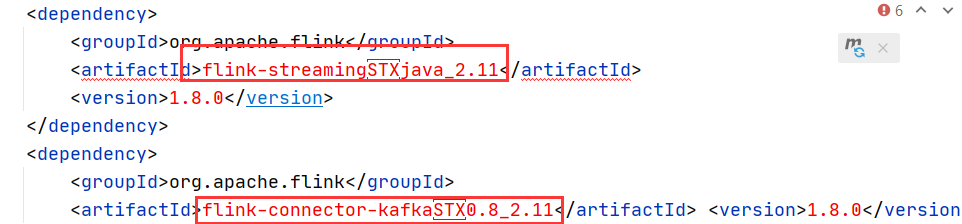
**1.Intellij导入包出错**

**问题描述：**Maven点击install后，报错

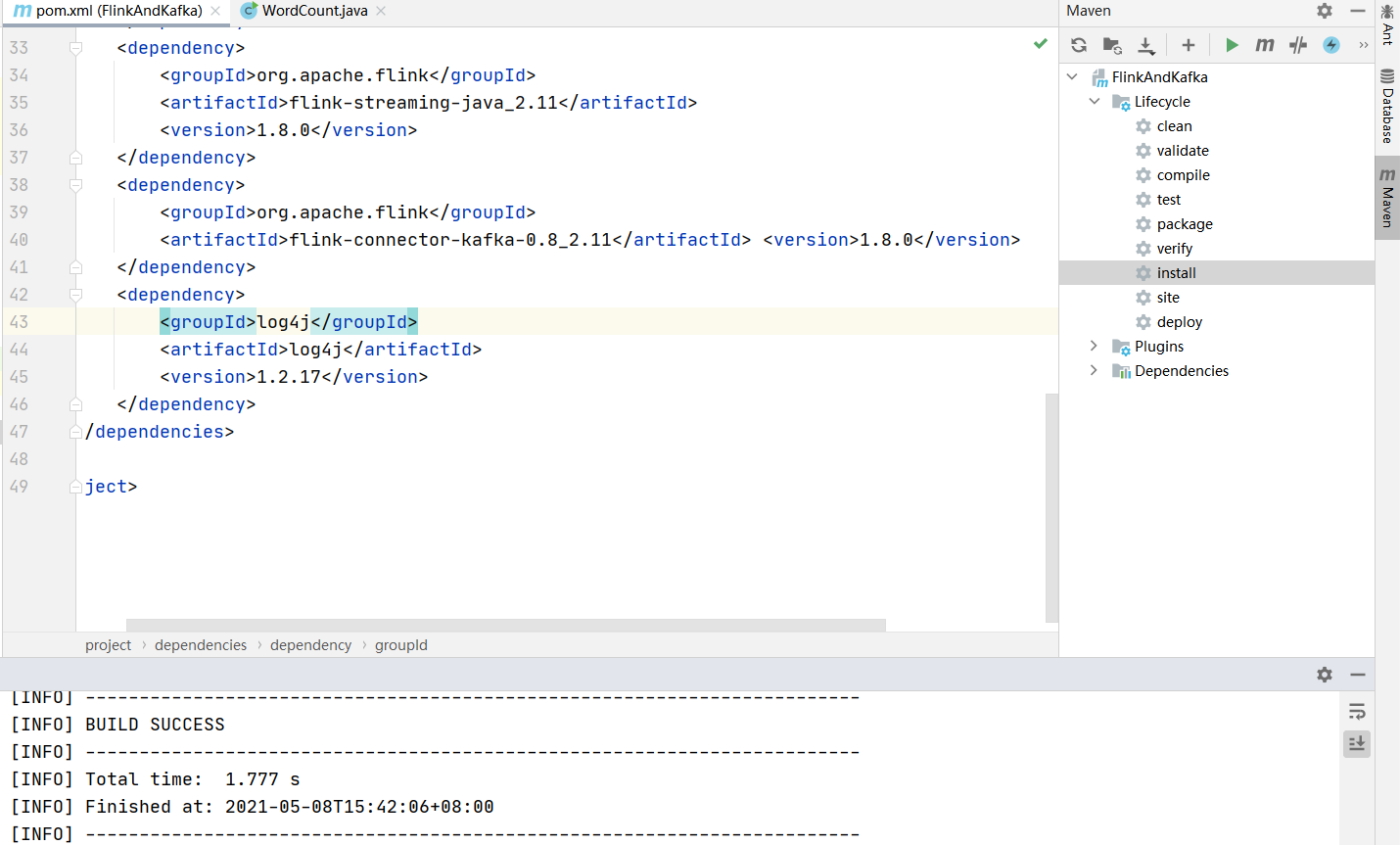


**图-1 install报错**

**错误分析：**Process terminated，说明install过程有问题，查看pom.xml中的依赖包名后发现如下，原因是从pdf文件复制出错，修改后错误消失



**图-2 出错包名**

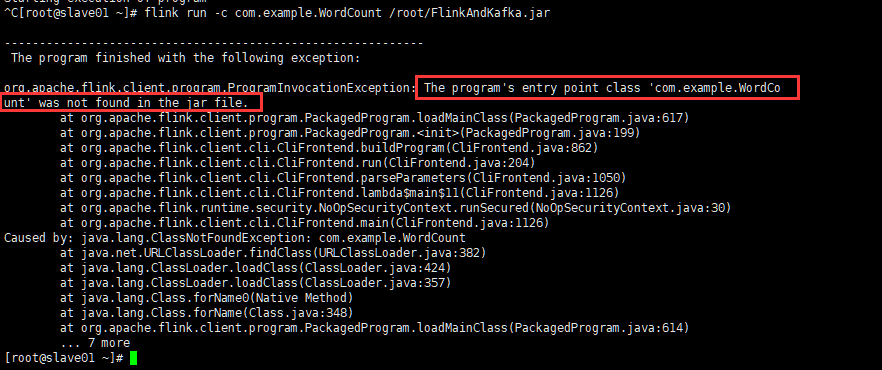


**图-3 修改后程序**

**问题思考：**复制过程出错，仔细检查后发现问题。暴露出做实验时应该细致，直接复制不可取。

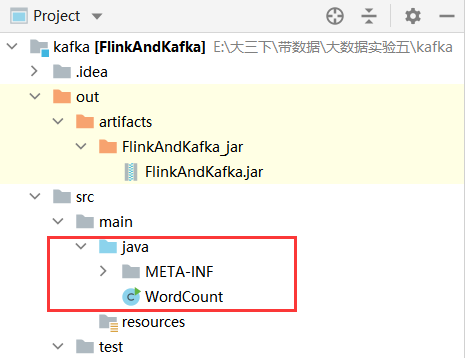
**2.启动类出错**

**问题描述：**启动WordCount报错



**图-4 启动报错**

**错误分析：**入口类出错，程序入口类应该直接为WordCount，修改后启动成功



**图-5 项目index**



**图-6 启动成功**

**问题思考：**要仔细思考每个指令的写法和语法，根据报错认真修改

**四.实验代码**

|  |
| --- |
| **import** org.apache.flink.api.common.functions.FlatMapFunction;  **import** org.apache.flink.api.common.serialization.SimpleStringSchema;  **import** org.apache.flink.api.java.tuple.Tuple2;  **import** org.apache.flink.streaming.api.datastream.DataStream;  **import** org.apache.flink.streaming.api.environment.StreamExecutionEnvironment;  **import** org.apache.flink.streaming.connectors.kafka.FlinkKafkaConsumer08;  **import** org.apache.flink.util.Collector;  **import** java.util.Properties;    **public** **class** WordCount {  **public** **static** **void** main(String[] args) **throws** Exception  {  */\*获取 Flink 运行环境\*/*  StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment();  */\*配置 Kafka 连接属性\*/*  Properties properties = **new** Properties();  properties.setProperty("bootstrap.servers", "master:9092");  properties.setProperty("zookeeper.connect", "master:2181");  properties.setProperty("group.id", "1");  FlinkKafkaConsumer08<String> myconsumer = **new** FlinkKafkaConsumer08<>("test", **new** SimpleStringSchema(), properties);  */\*默认消费策略\*/*  myconsumer.setStartFromGroupOffsets();  DataStream<String> dataStream = env.addSource(myconsumer);  DataStream<Tuple2<String, Integer>> result = dataStream.flatMap(**new** MyFlatMapper()).keyBy(0).sum(1);  result.print().setParallelism(3);  env.execute();  }  **public** **static** **class** MyFlatMapper **implements** FlatMapFunction<String, Tuple2<String, Integer>> {  @Override  **public** **void** flatMap(String s, Collector<Tuple2<String, Integer>> out) **throws** Exception {  */\*按空格分词\*/*  String[] words = s.split(" ");  **for** (String word : words) {  out.collect(**new** Tuple2<>(word, 1));  }  }  }  } |