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

一、 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）

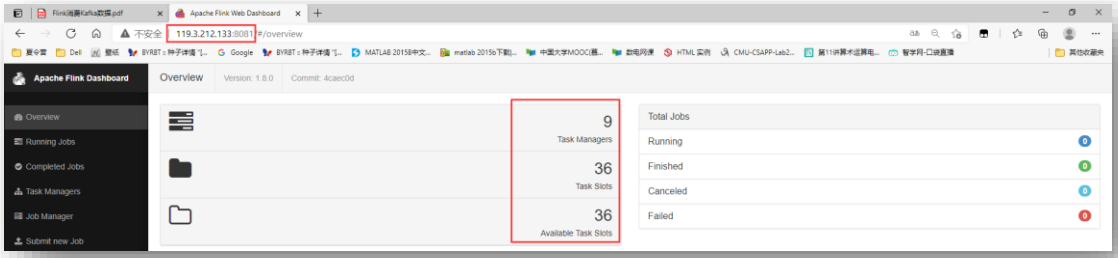
【结果截图 1】Zookeeper 安装验证

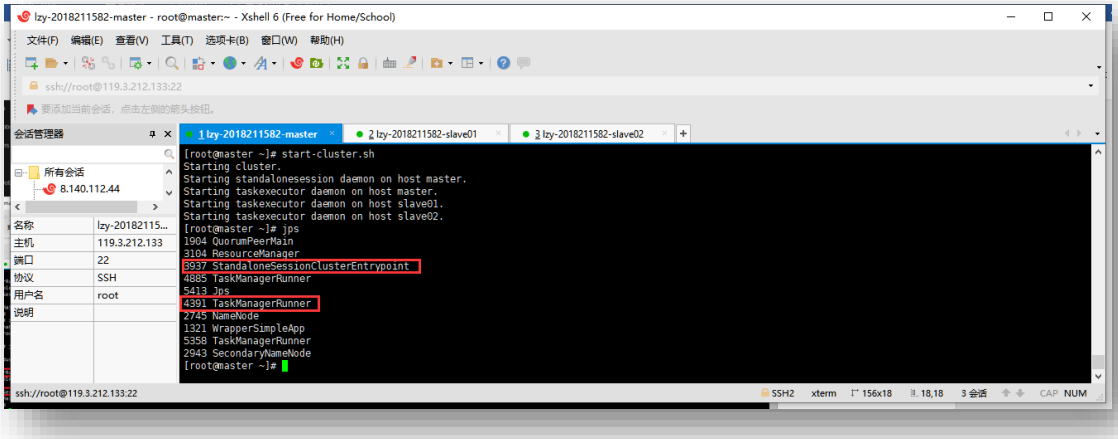
```
[root@slave01 ~]# zkServer.sh status
ZooKeeper JMX enabled by default
Using config: /root/zookeeper-3.4.12/bin/../conf/zoo.cfg
Mode: leader
[root@slave01 ~]#
```

```
[root@master ~]# zkServer.sh status
ZooKeeper JMX enabled by default
Using config: /root/zookeeper-3.4.12/bin/../conf/zoo.cfg
Mode: follower
[root@master ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.40  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::f816:3eff:fea3:d3d0  prefixlen 64  scopeid 0x20<link>
        ether fa:16:3e:a3:d3:d0  txqueuelen 1000  (Ethernet)
        RX packets 258119  bytes 350347751 (334.1 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 109799  bytes 129049344 (123.0 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 20  bytes 1508 (1.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 20  bytes 1508 (1.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[root@master ~]#
```

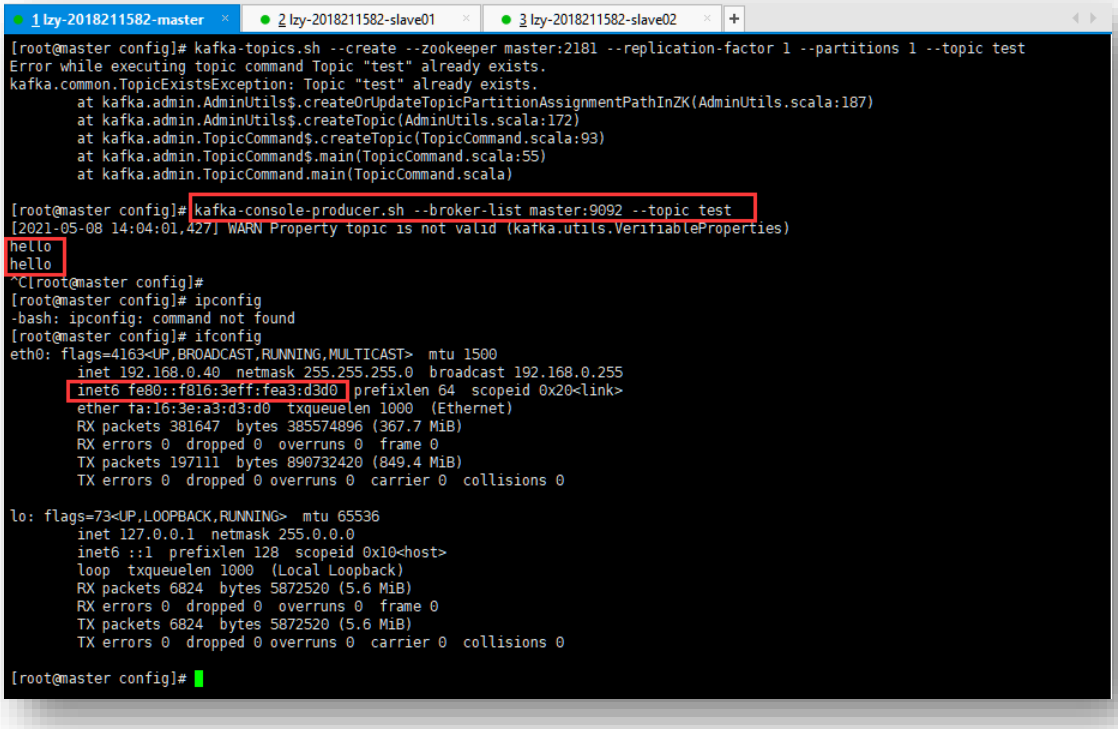
```
[root@slave02 ~]# zkServer.sh status
ZooKeeper JMX enabled by default
Using config: /root/zookeeper-3.4.12/bin/../conf/zoo.cfg
Mode: follower
[root@slave02 ~]#
```

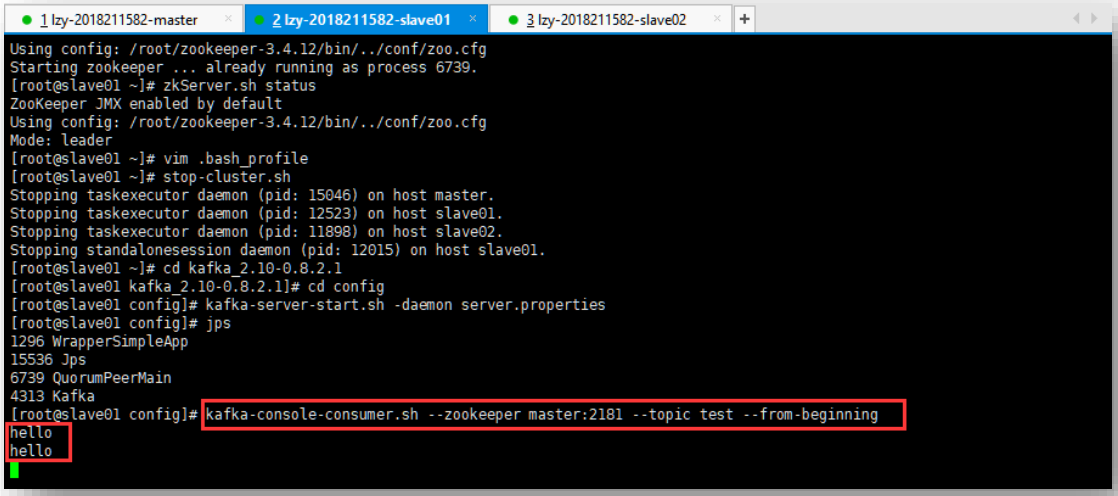
【结果截图 2】Flink 安装验证



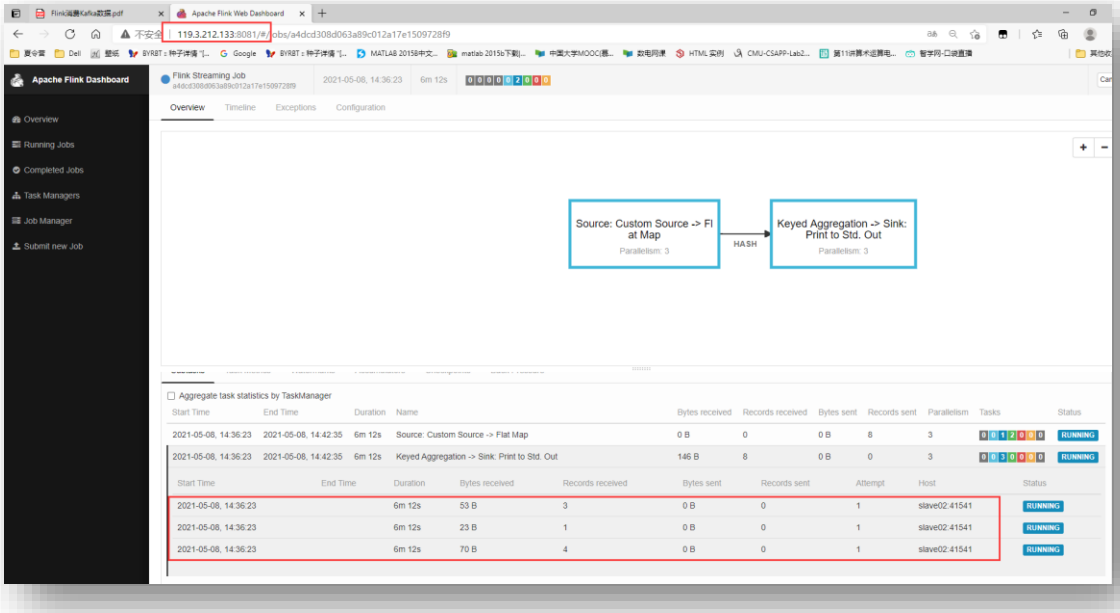


【结果截图 3】Kafka 对话验证

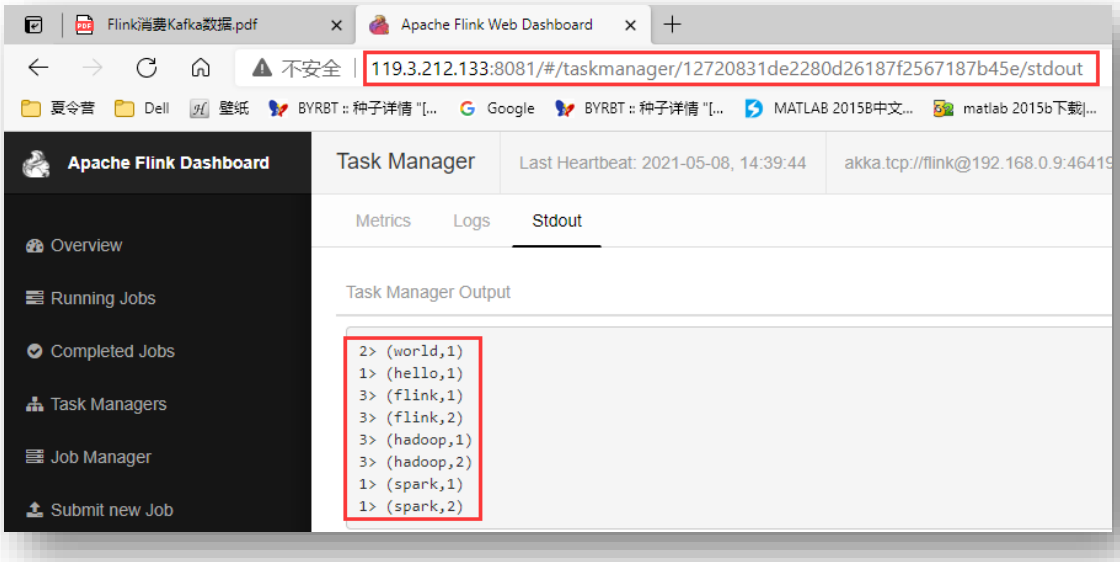


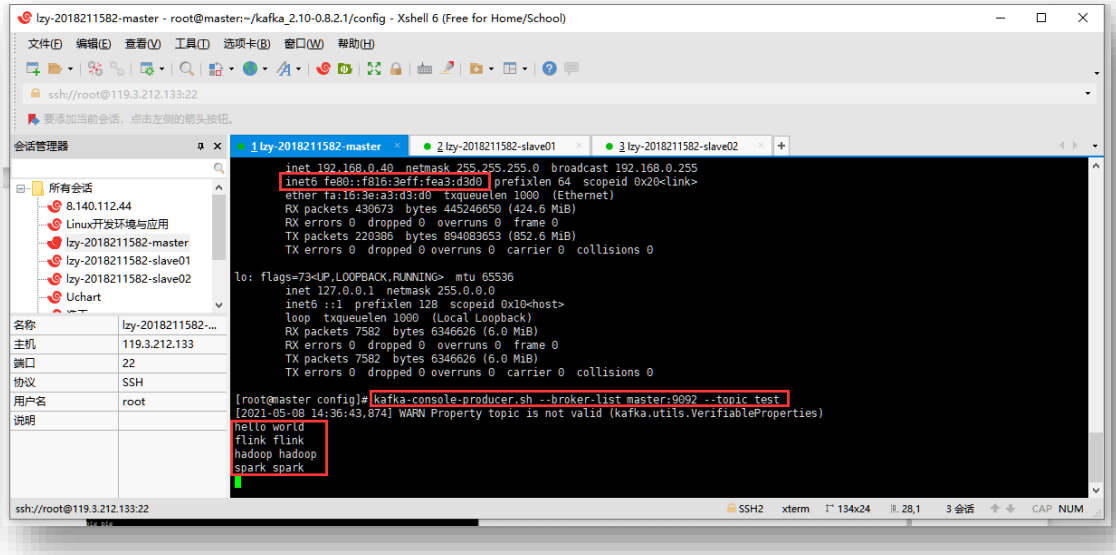


【结果截图 4】查询输出结果节点



【结果截图 5】单词计数





二、 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

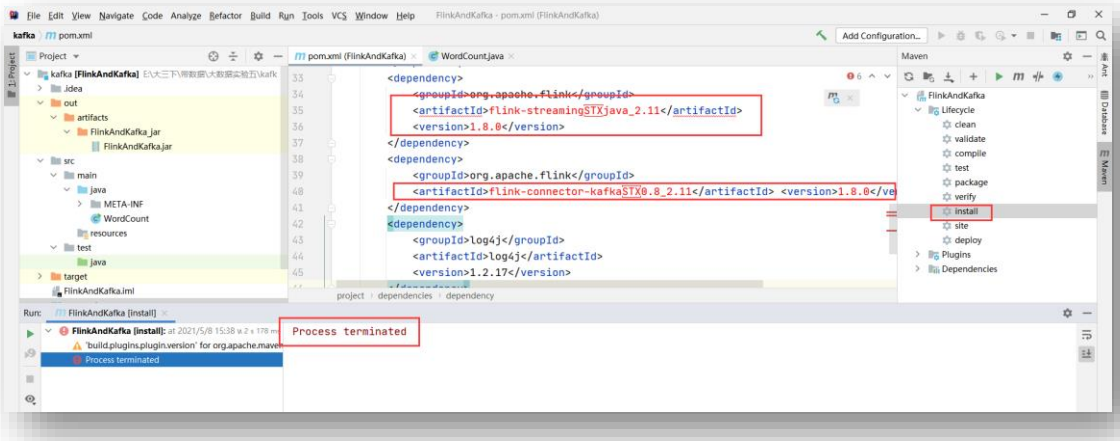


图-1 install 报错

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

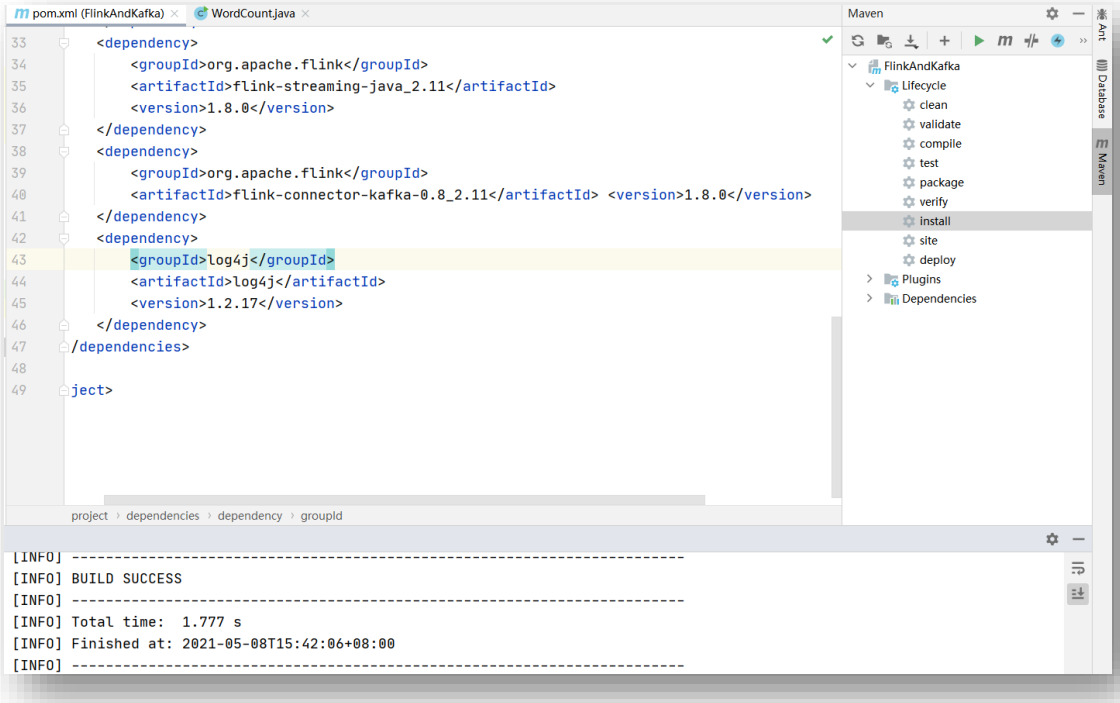


图-3 修改后程序

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

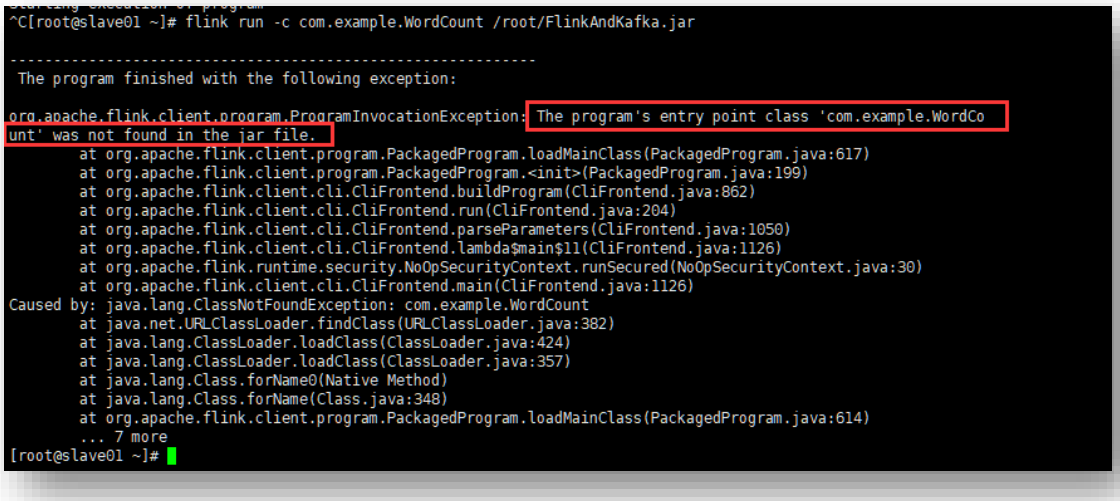


图-4 启动报错

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

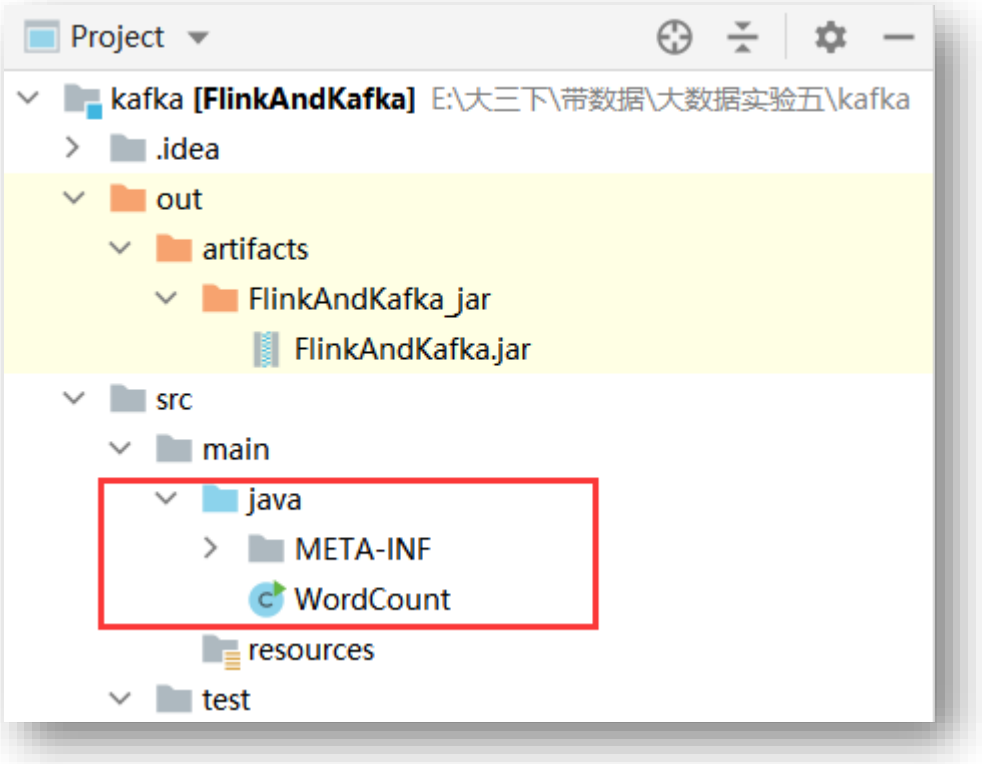


图-5 项目 index

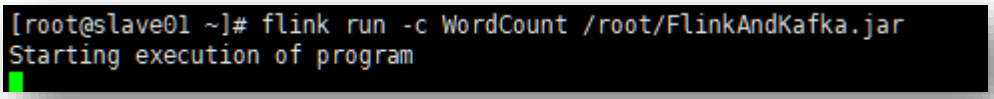


图-6 启动成功

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.StreamExecutionEnvironme
nt;

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.getExecu
tionEnvironment();

        /*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, Tupl
e2<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));

            }

        }

    }

}
```