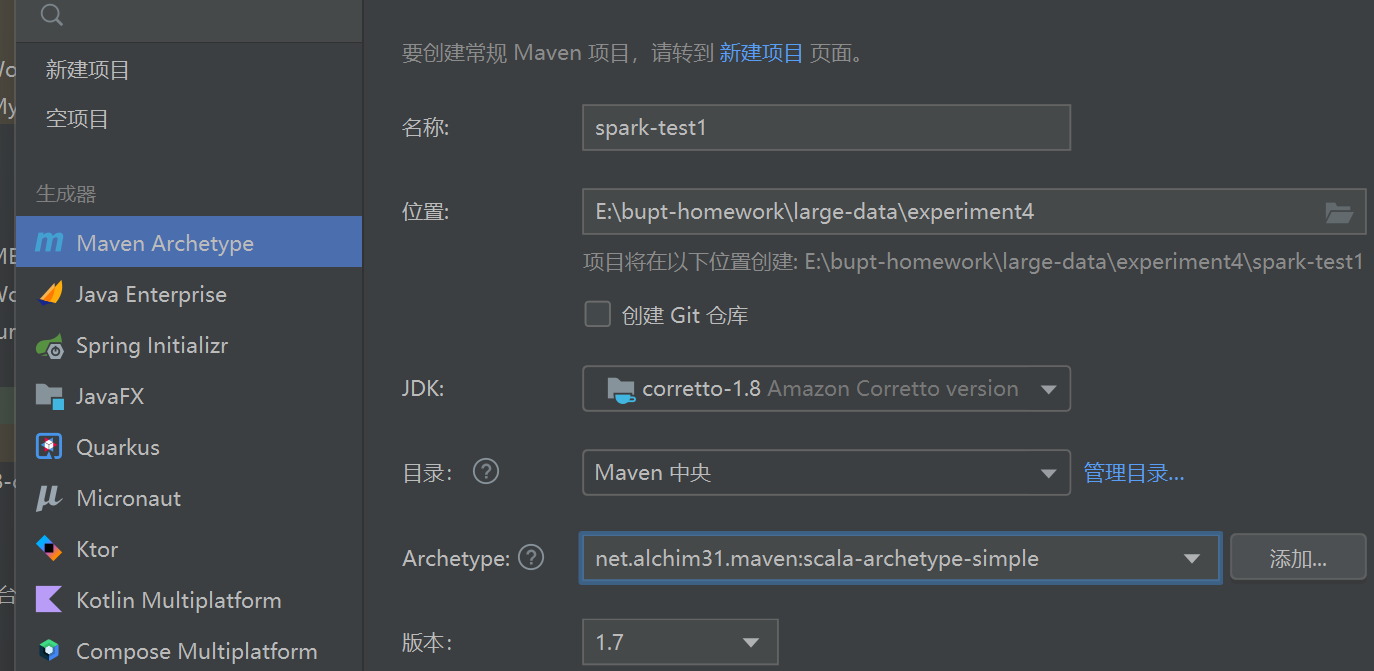
到spark版本为3.5.1，scala版本为2.13.8。

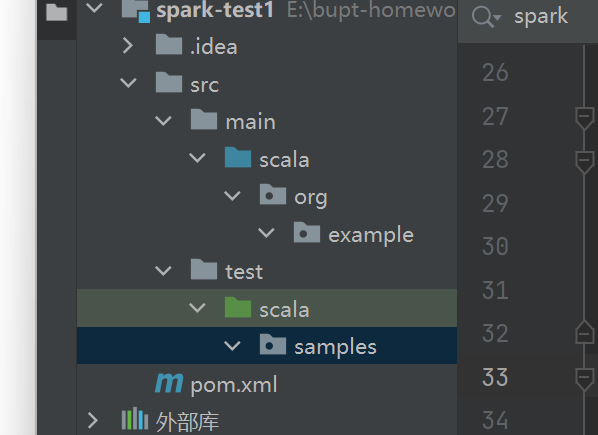
1. 编写程序

先在IDEA安装Scala插件

创建项目



删除test文件夹和main文件夹下的文件

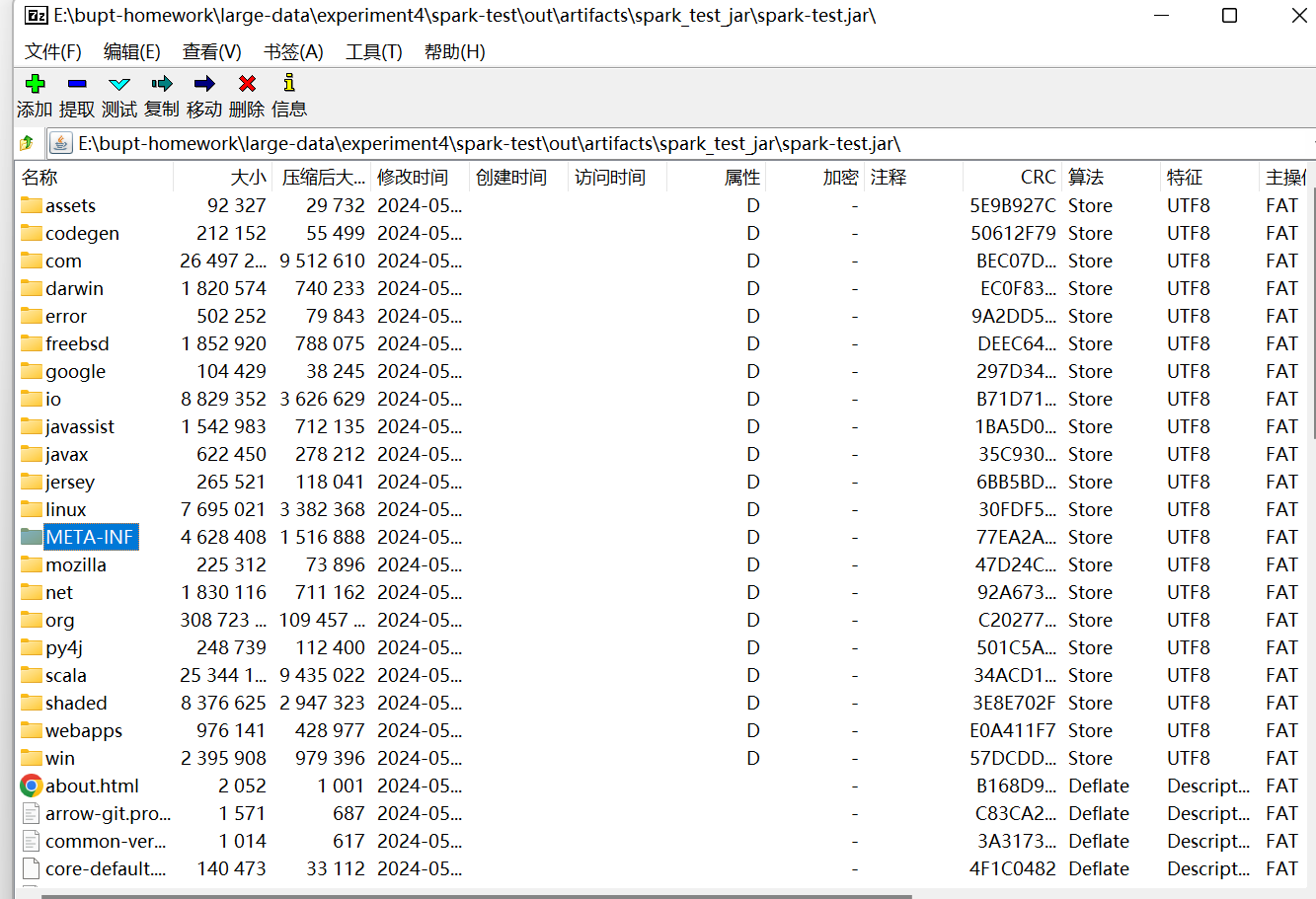


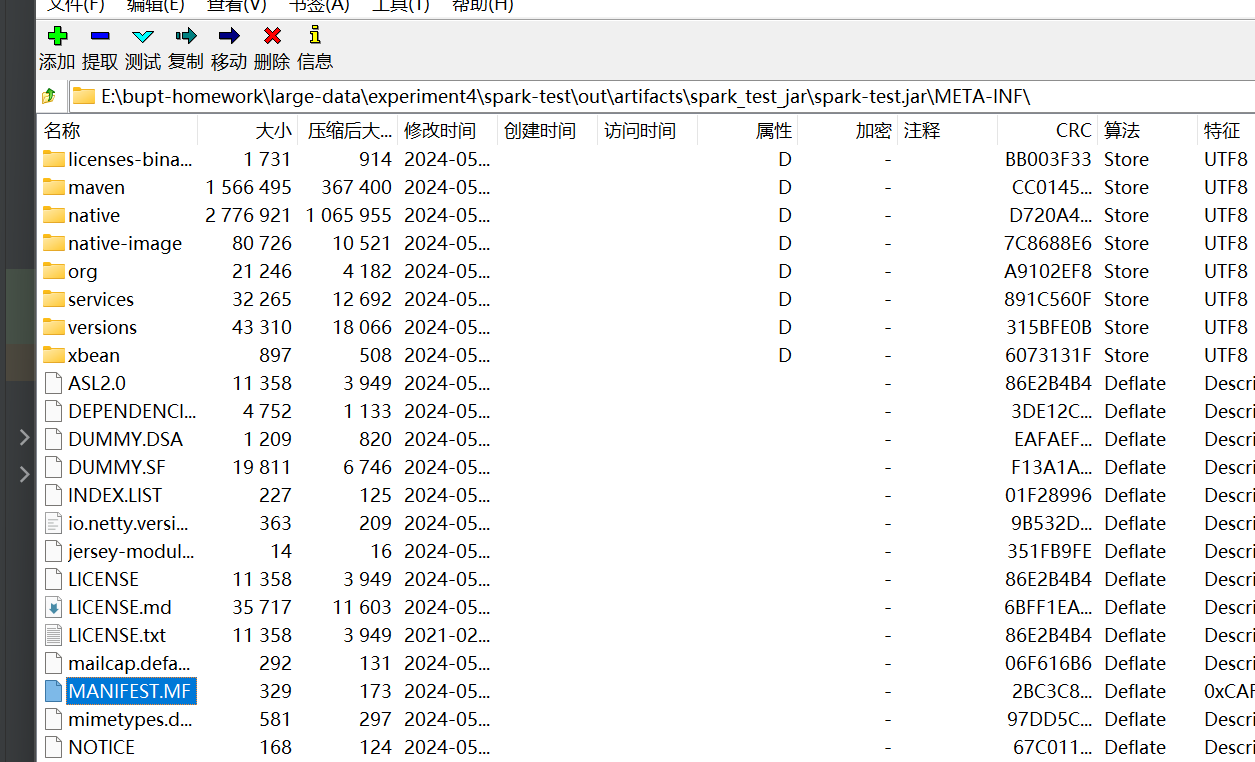
构建项目

文件 -> 项目结构 -> 工件

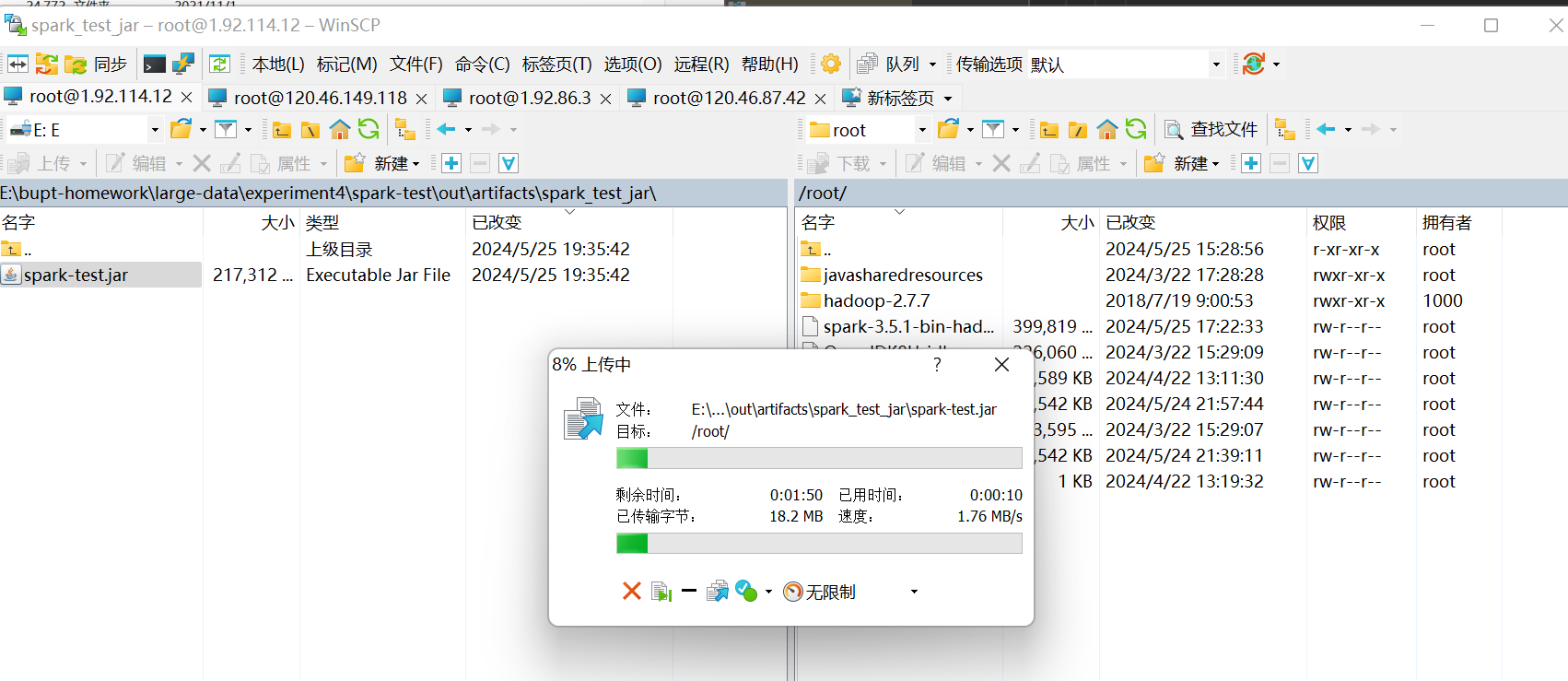
构建 -> 构建工件

删除jar包中的META-INF中的META-INF.MF

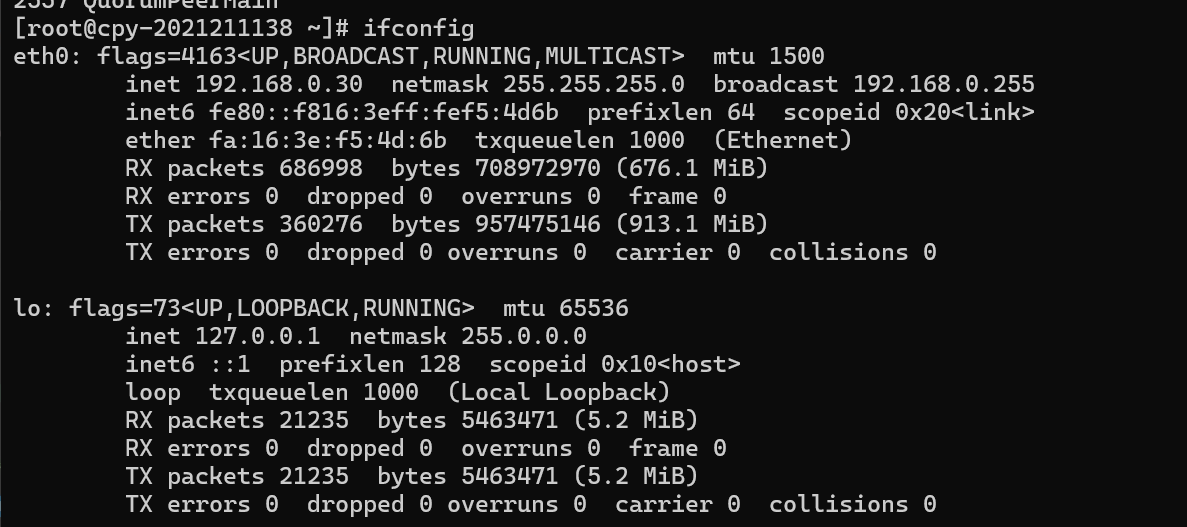




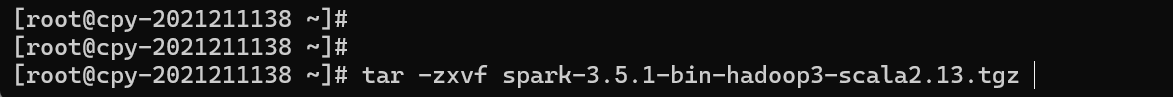
将spark-test.jar包传到/root下面

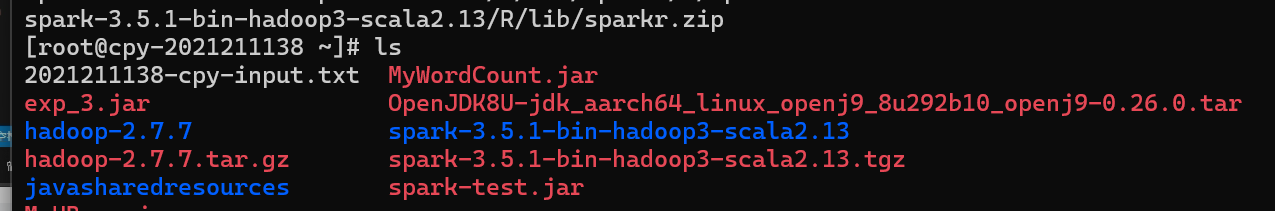


环境配置



解压spark安装包





修改用户变量

cd ~

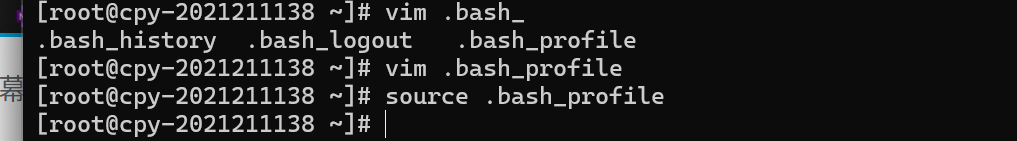
vim .bash\_profile

添加下面这四行



使刚才的修改生效：

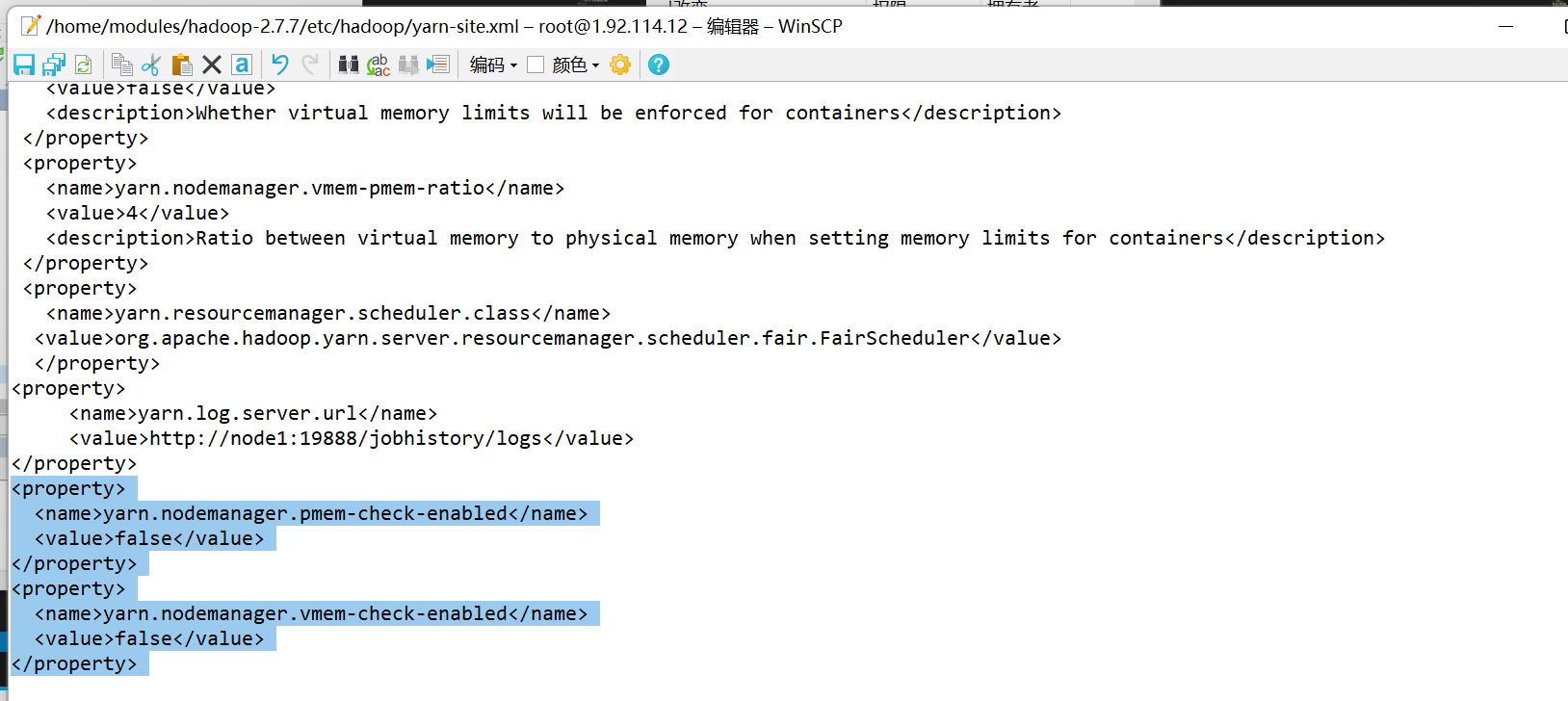
source .bash\_profile



修改hadoop的yarn配置：

cd ../home/modules/hadoop-2.7.7/etc/hadoop

进入上面这个路径，添加下面的配置



先关闭hadoop

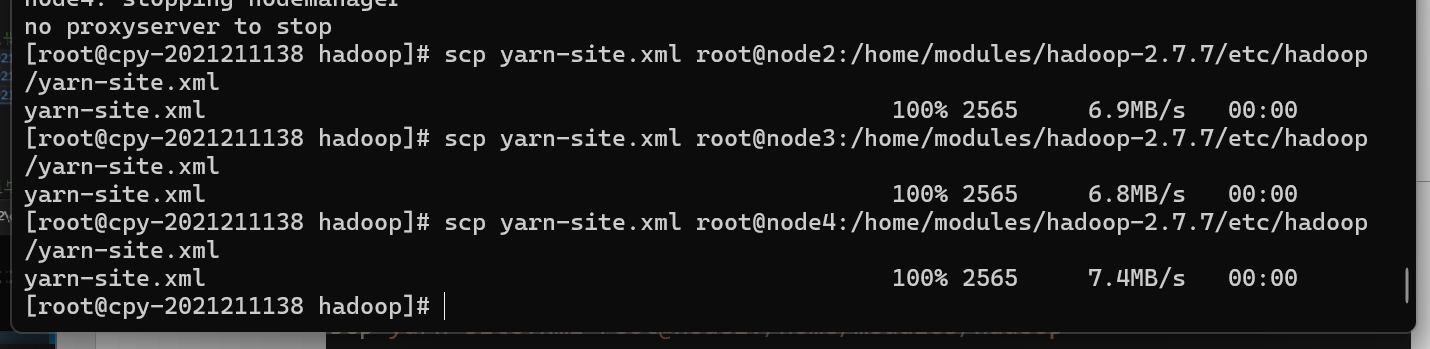
stop-all.sh

将yarn-site.xml分发给其他节点

scp yarn-site.xml root@node2:/home/modules/hadoop-2.7.7/etc/hadoop/yarn-site.xml

scp yarn-site.xml root@node3:/home/modules/hadoop-2.7.7/etc/hadoop/yarn-site.xml

scp yarn-site.xml root@node4:/home/modules/hadoop-2.7.7/etc/hadoop/yarn-site.xml



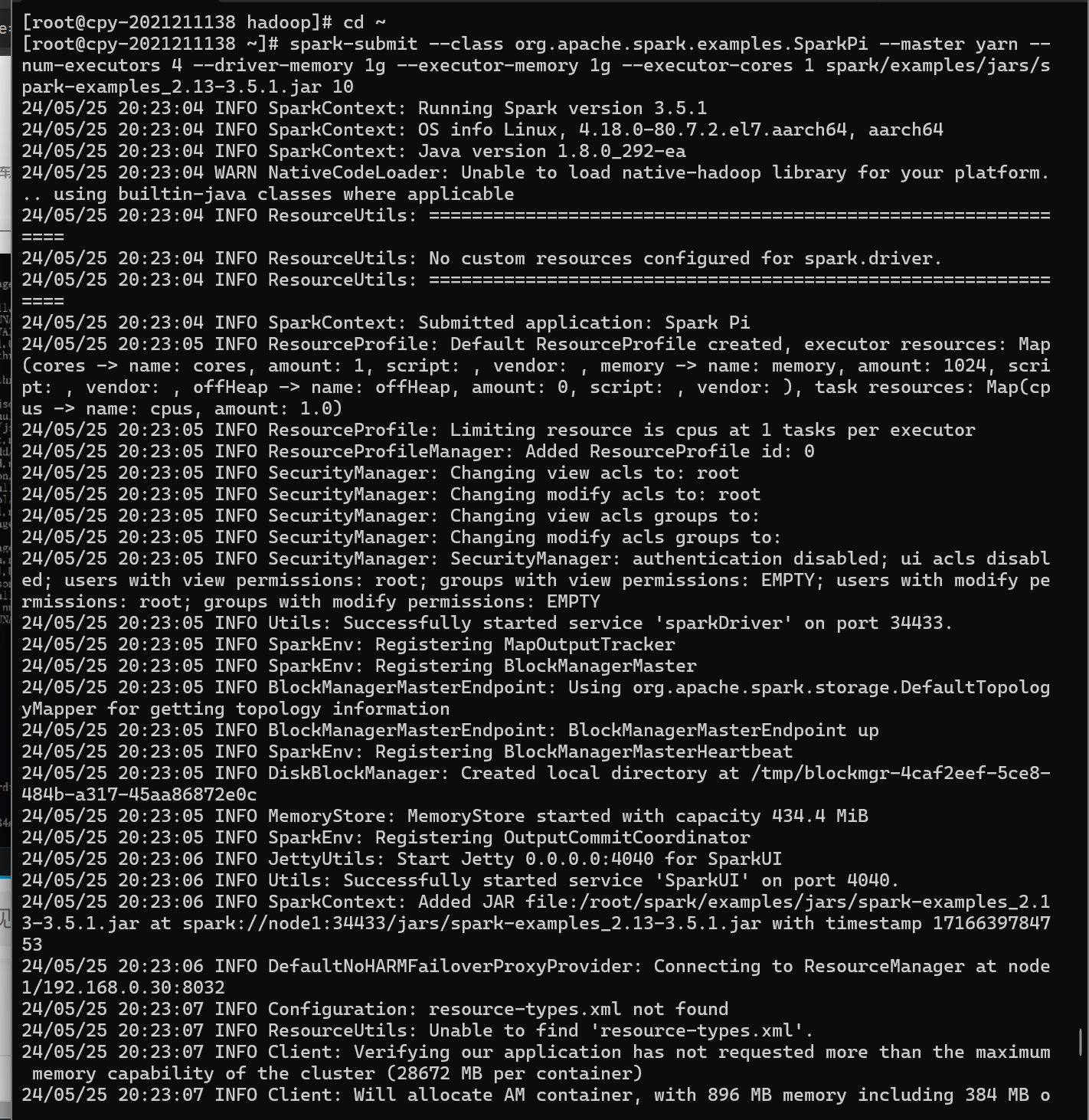
查看spark是否安装成功：

首先进入用户目录下：

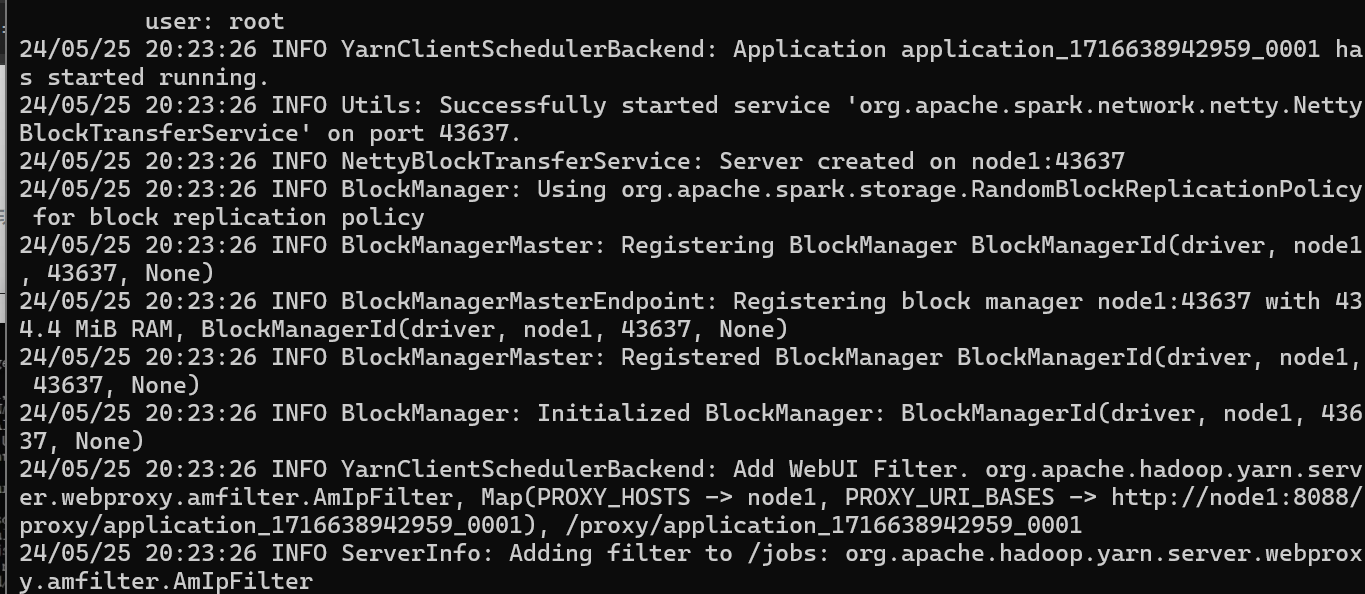
cd ~

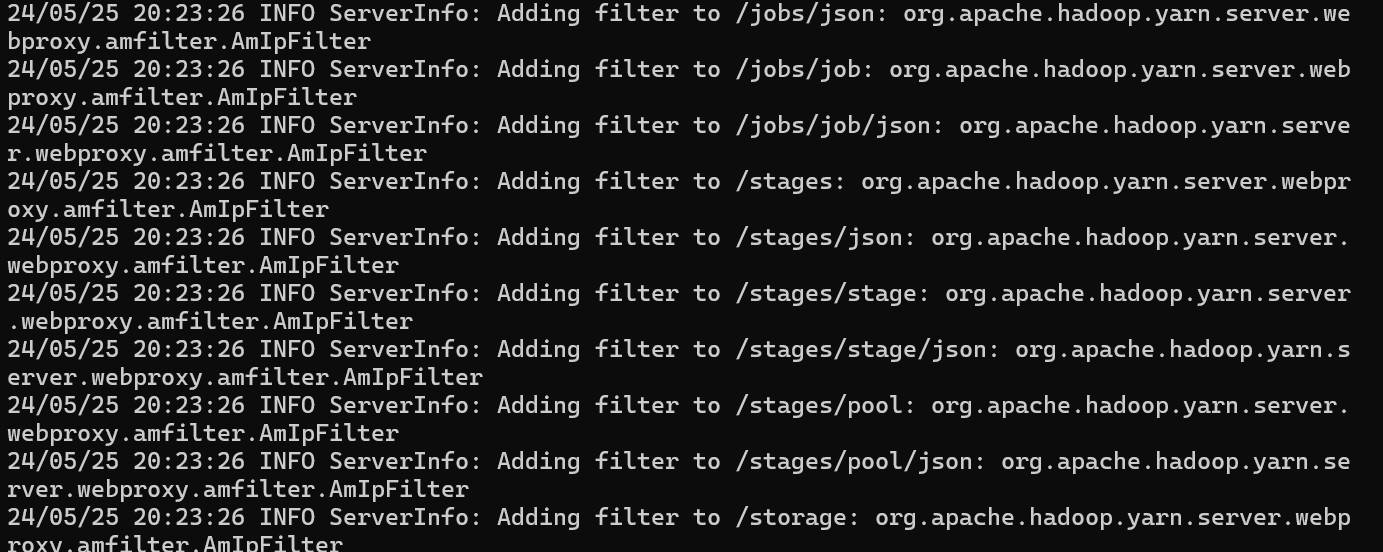
输入：

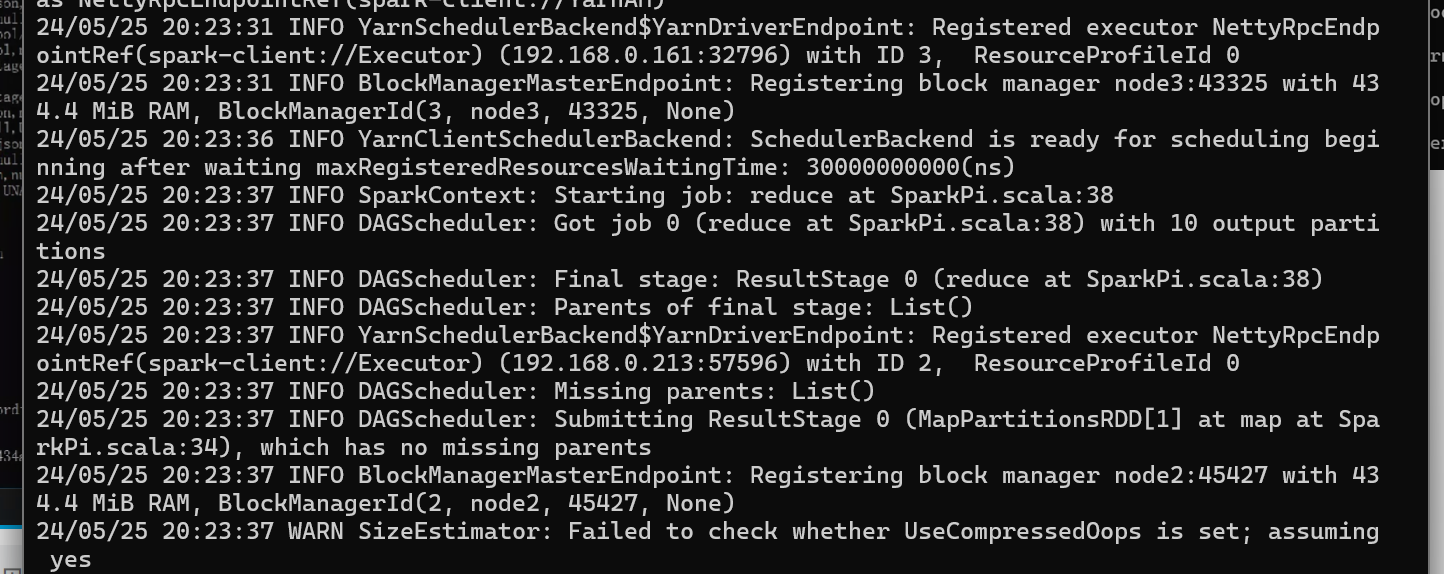
spark-submit --class org.apache.spark.examples.SparkPi --master yarn --num-executors 4 --driver-memory 1g --executor-memory 1g --executor-cores 1 spark/examples/jars/spark-examples\_2.13-3.5.1.jar 10

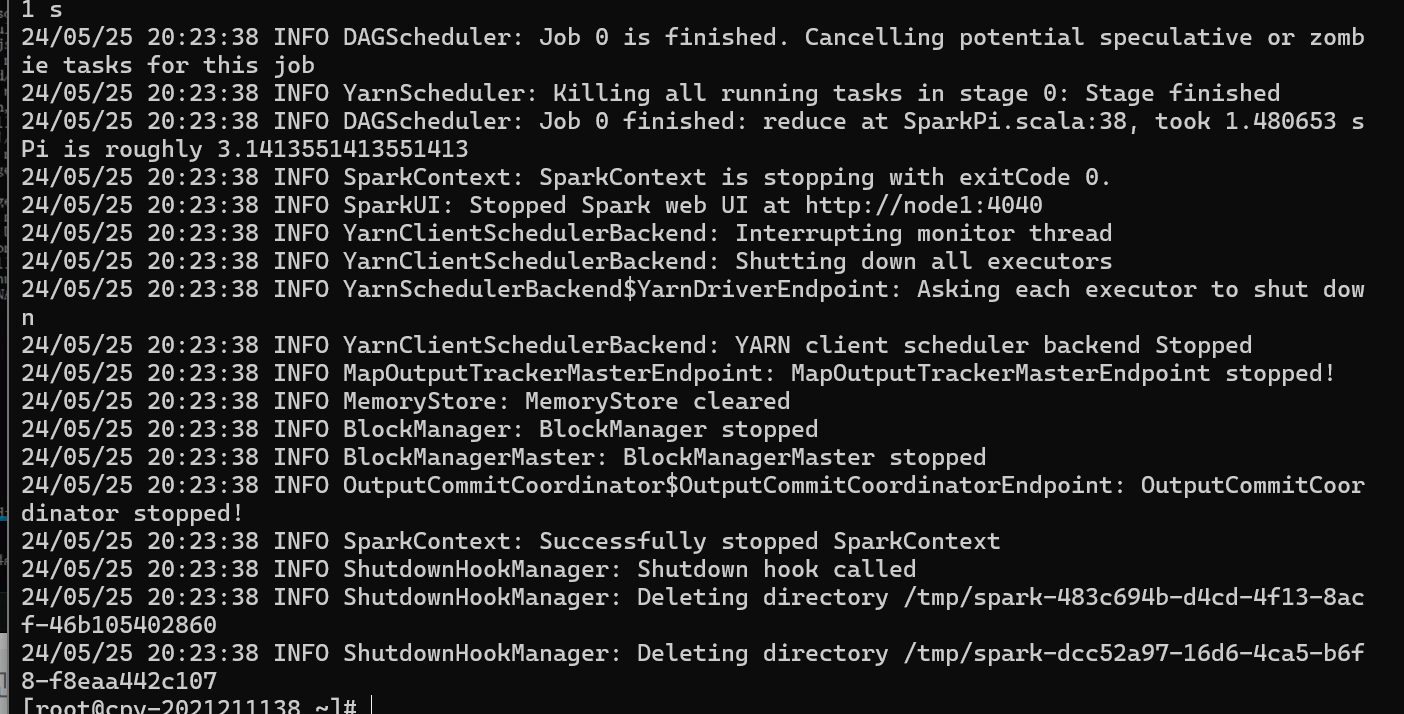


0

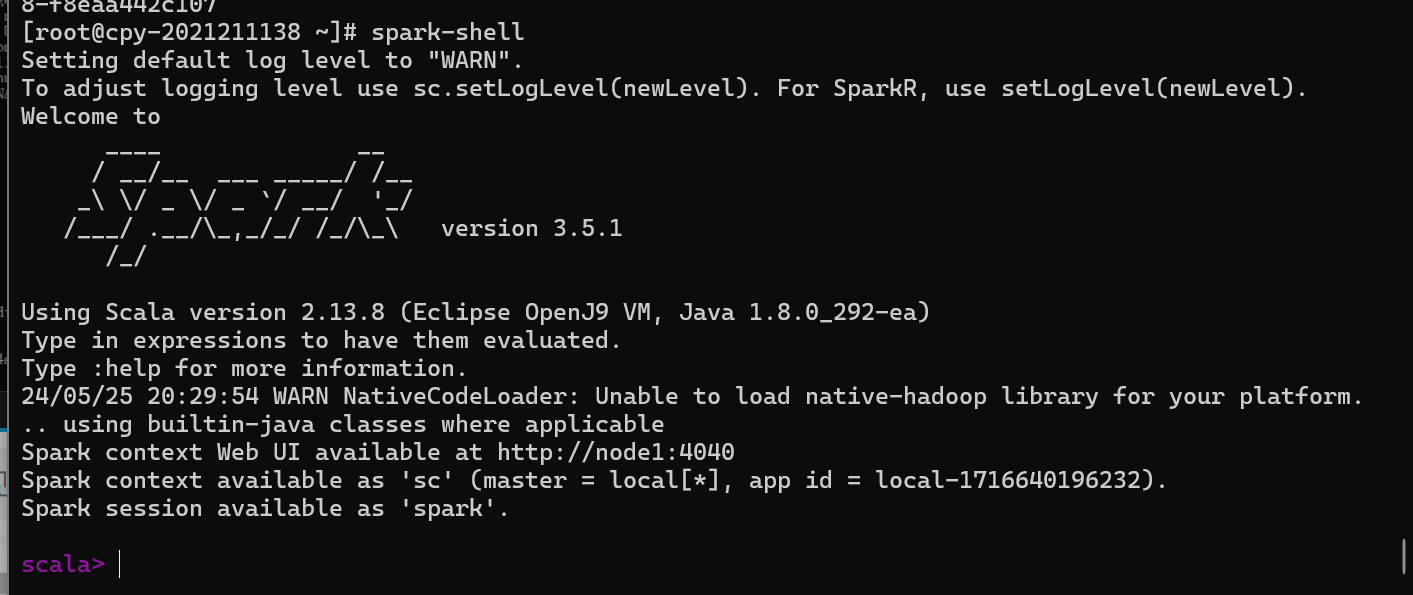








输入 spark-shell 查看版本信息



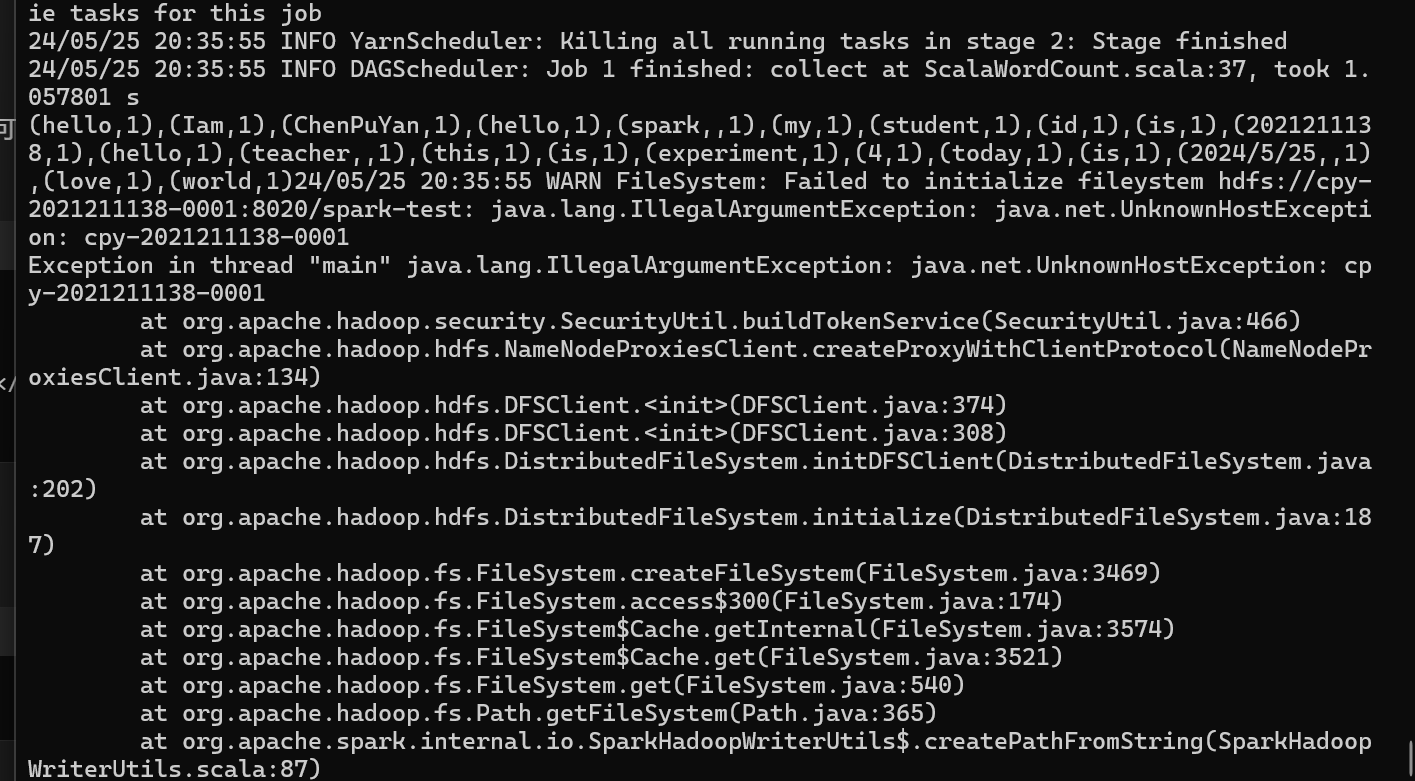
退出指令：

:quit

cd ~ 运行如下指令，

spark-submit --class org.example.ScalaWordCount --master yarn --num-executors 3 --driver-memory 1g --executor-memory 1g --executor-cores 1 spark-test.jar

报错：

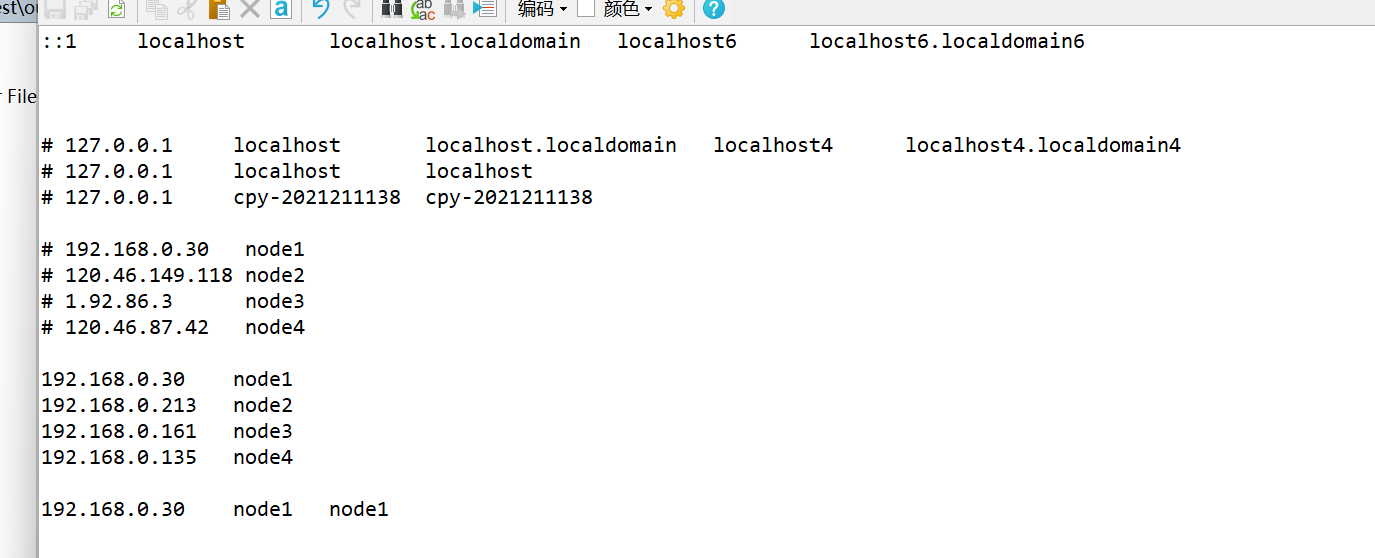


查看异常信息，主要错在：

java.lang.IllegalArgumentException: java.net.UnknownHostException: cpy-2021211138-0001

也就是说我的主机名不对。

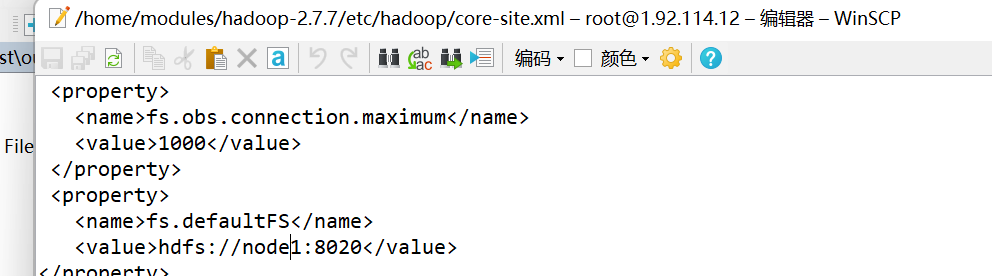
查看/etc/hosts文件：



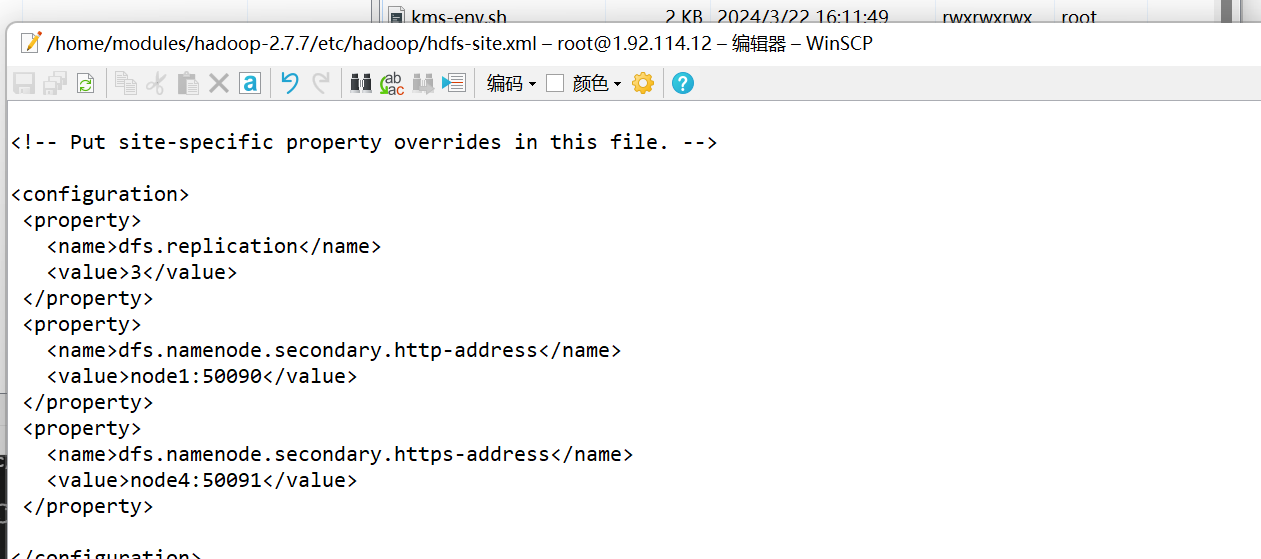
这里都是用node1、node2、node3、node4，并没有其他的主机名

但是我自己回想一下，好像有些地方确实改了主机名，于是一个一个排查：

hadoop下的core-site.xml中用的是node1，不是这个文件出错

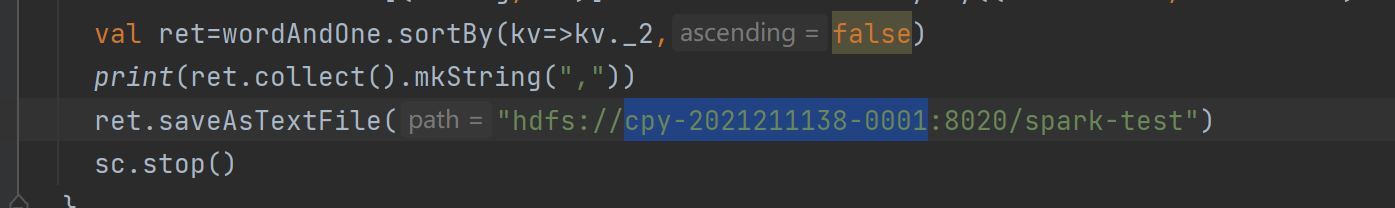


hdfs-site.xml用的也是node的编号



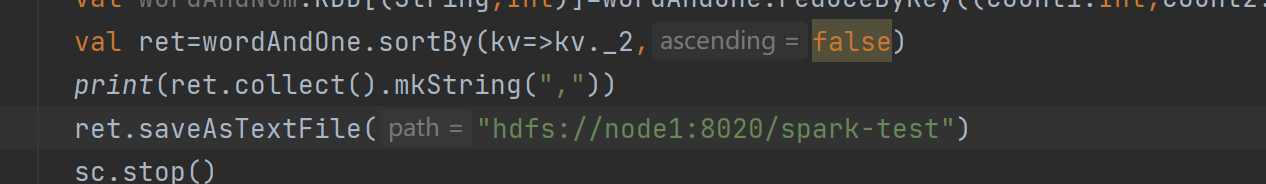
yarn-site.xml中也没有

找了半天，发现原来是代码错了：



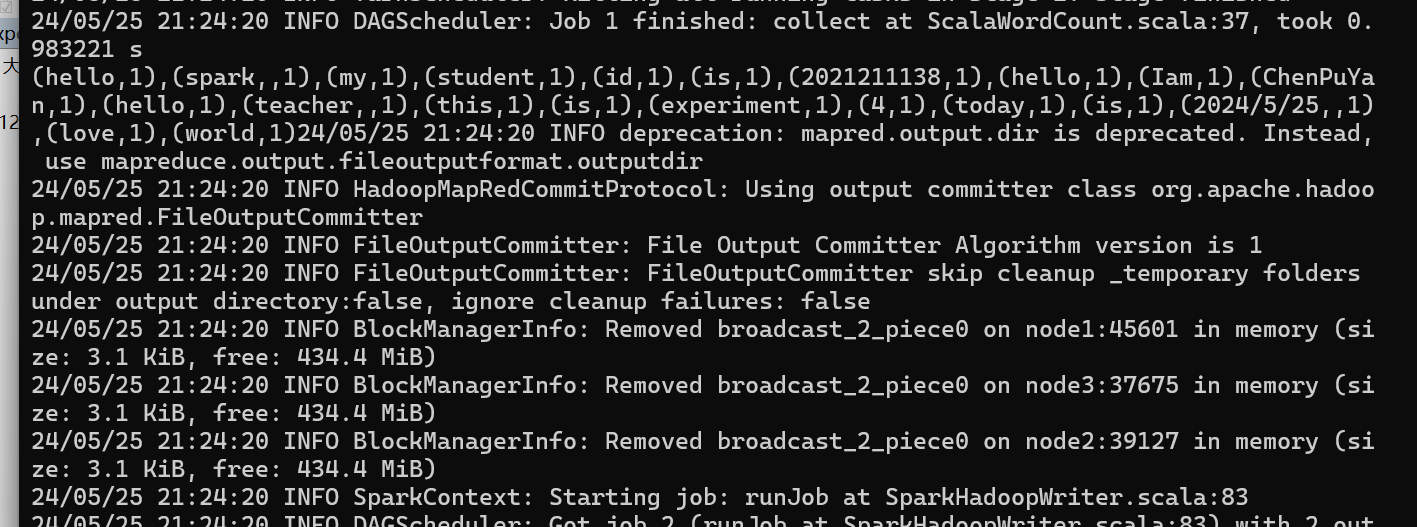
我应该在标蓝的这里写node1的

修改如下：

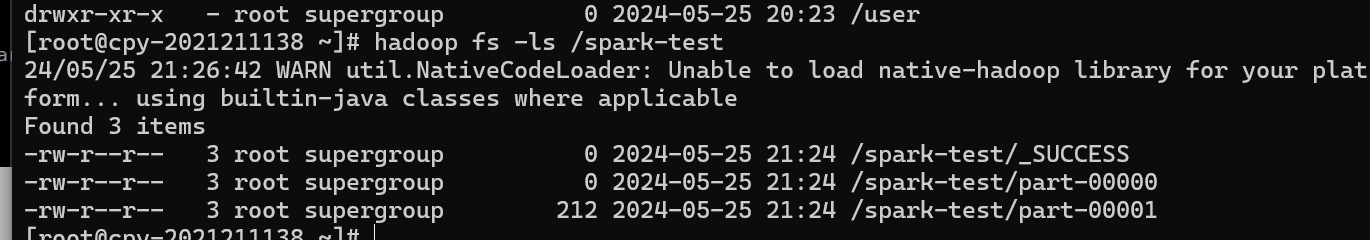


之后重新打包，重新上传

之后再运行一次就没有明显的报错了

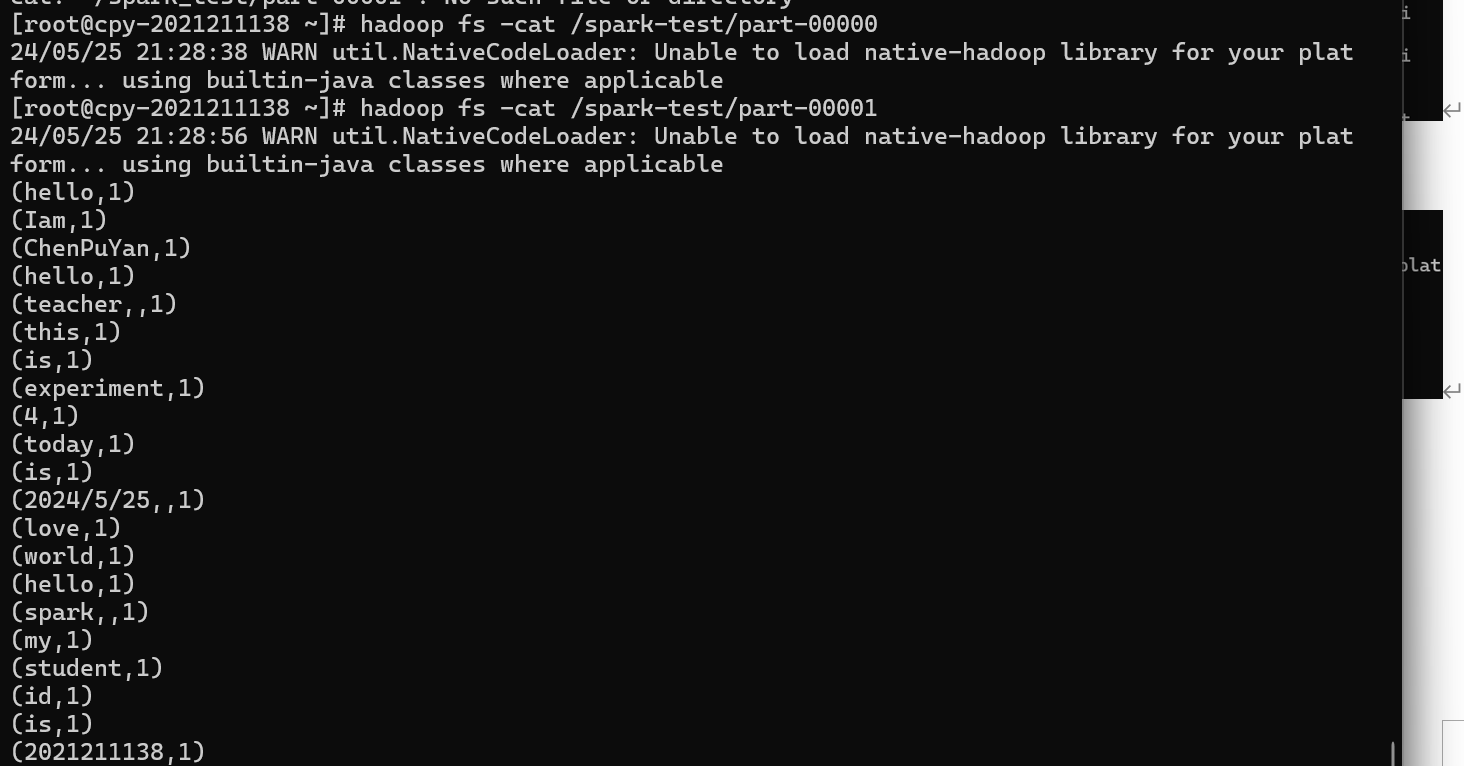


在hdfs上查看程序输出：



出现两个文件

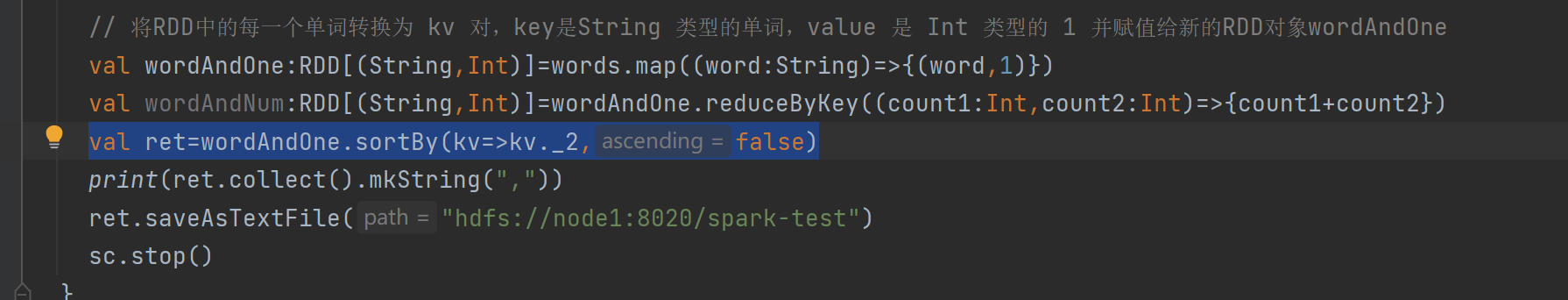
使用 hadoop fs -cat /spark-test/part-0000x来查看输出信息，得到如下内容：



但是这里只有一个文件有内容，并且我也只有两个文件，跟实验报告中的4个文件显然不同。

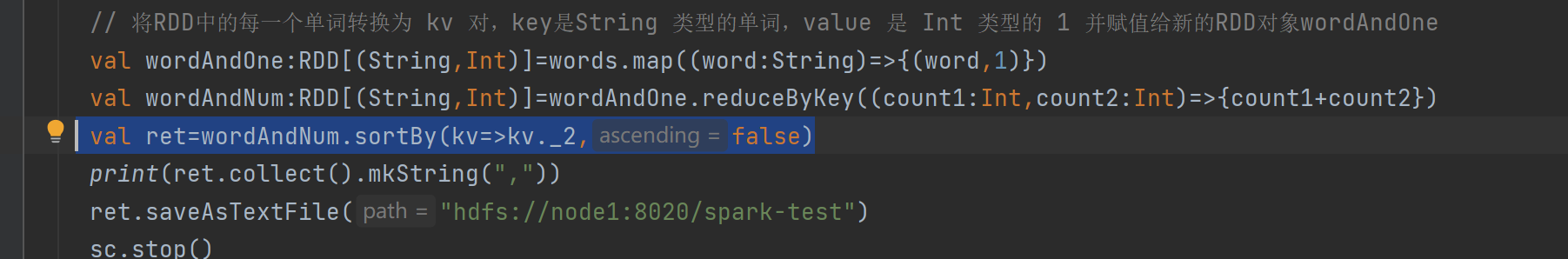
原因如下：





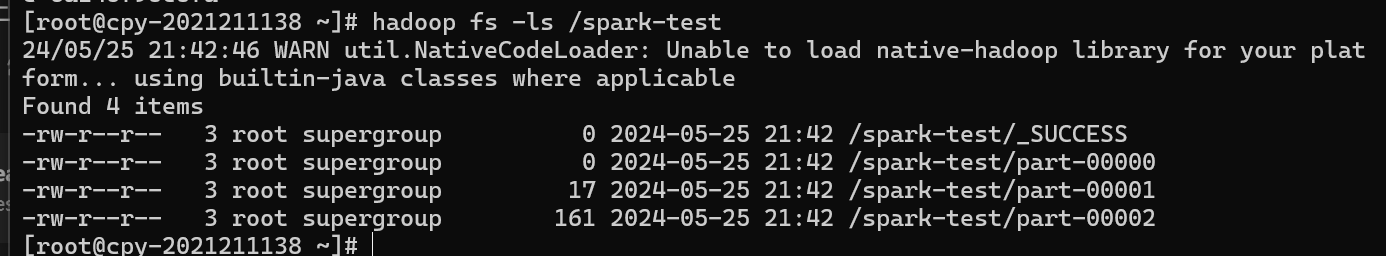
标蓝的地方写错了，应该是wordAndNum而不是wordAndOne。

修改后，如下：



重新打包，删除MANIFEST.FT文件，重新上传，并删除hadoop fs里面的/spark-test文件夹，

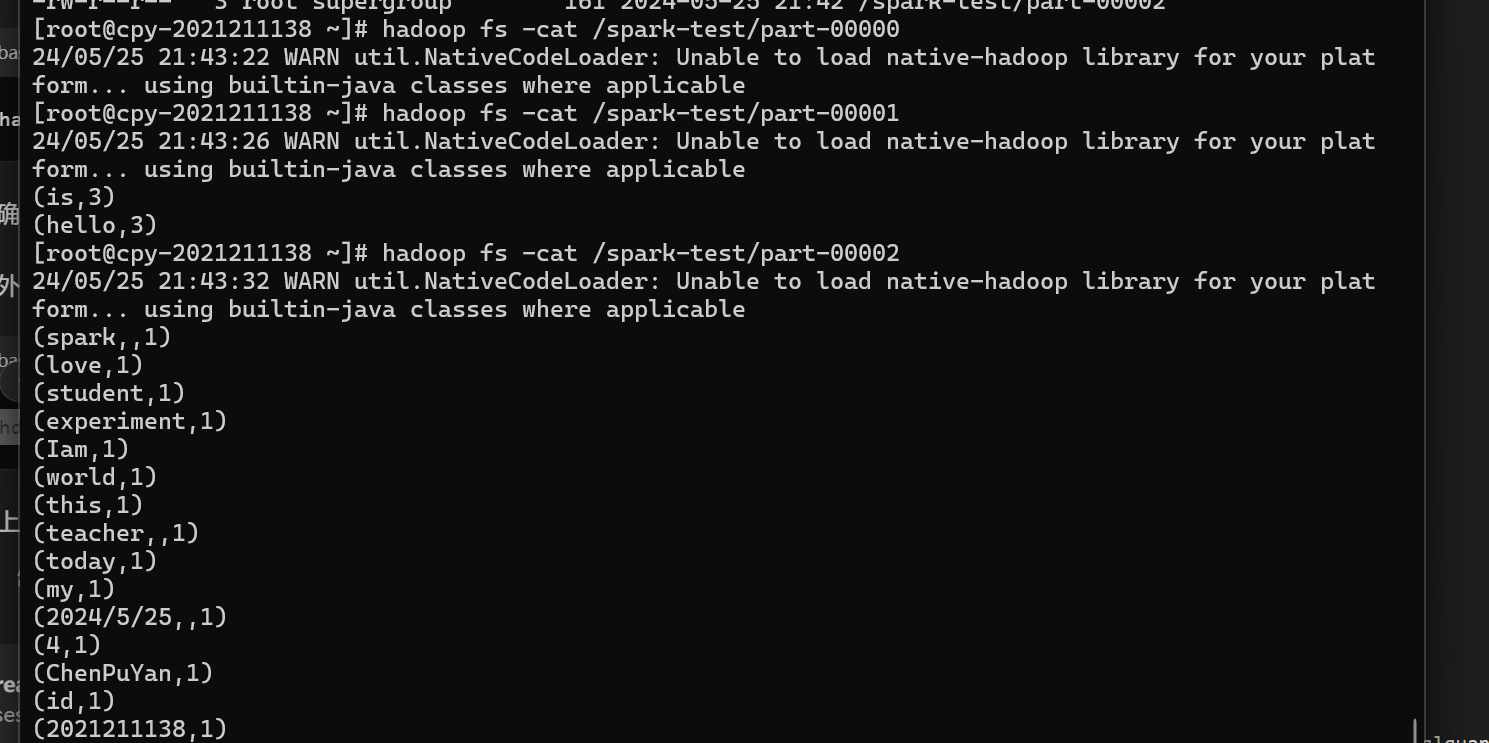
重新运行：



可以看到现在有了三个输出文件

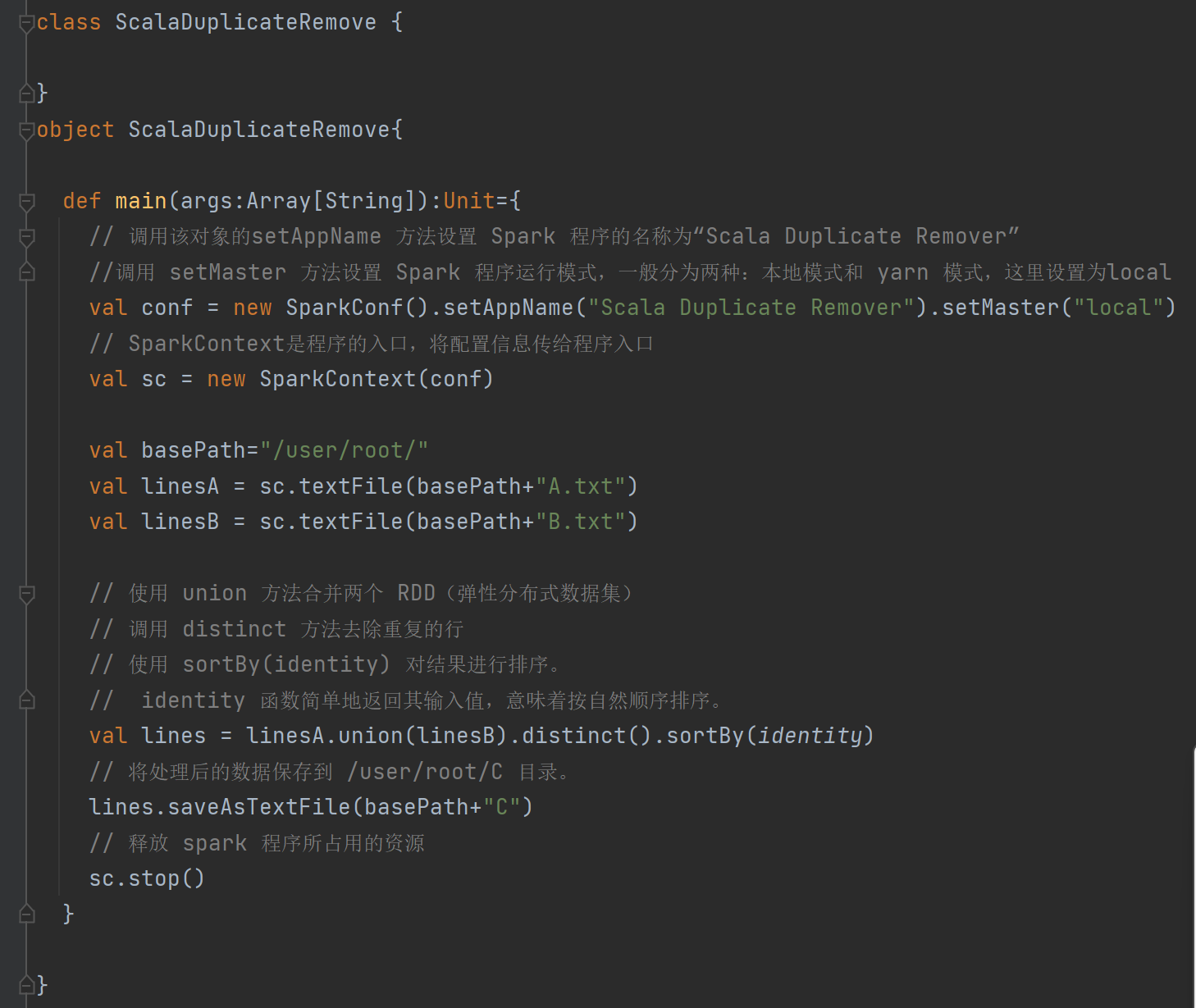
检擦三个输出文件

结果如下：



得到正确的结果。

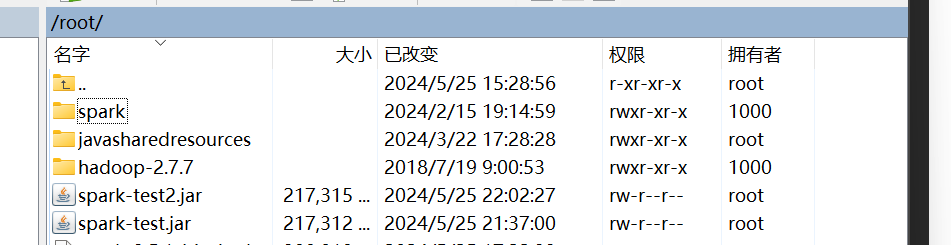
第二个实验



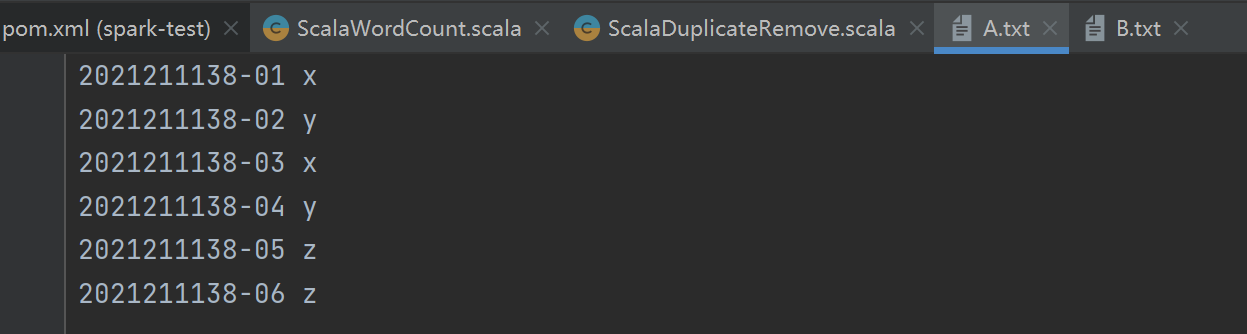
编写代码，构建工件，得到out包

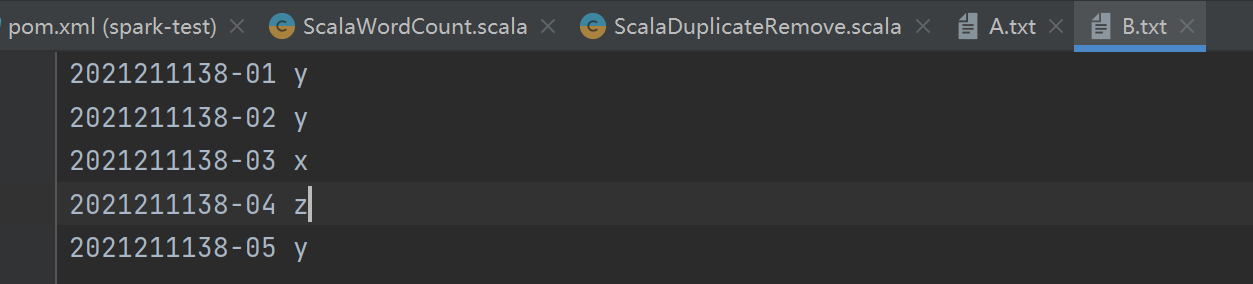
删除掉jar包中的MANIFEST.MF文件

用SCP传到node1主机上的/root文件夹下，并重命名为spark-test2.jar

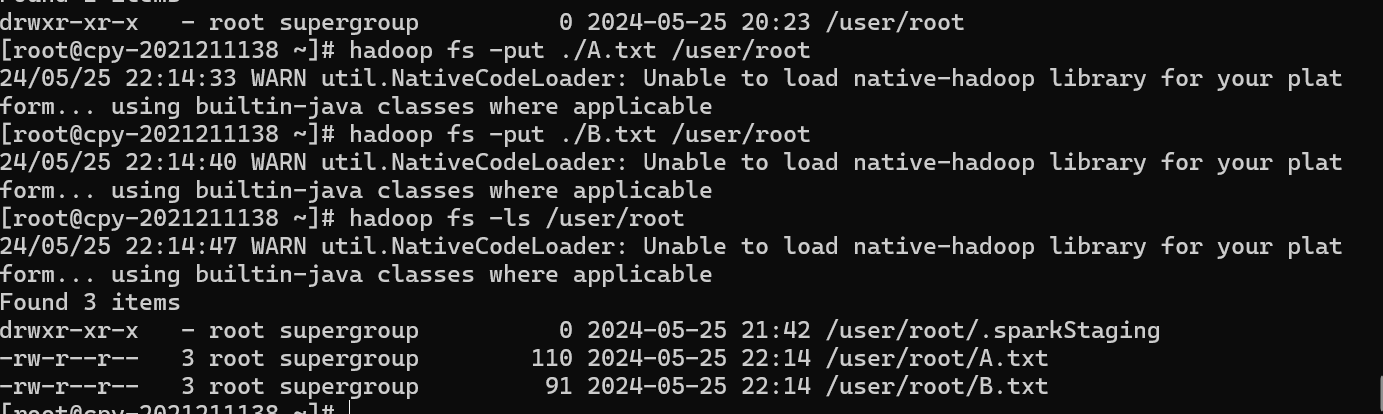


编写A.txt、B.txt





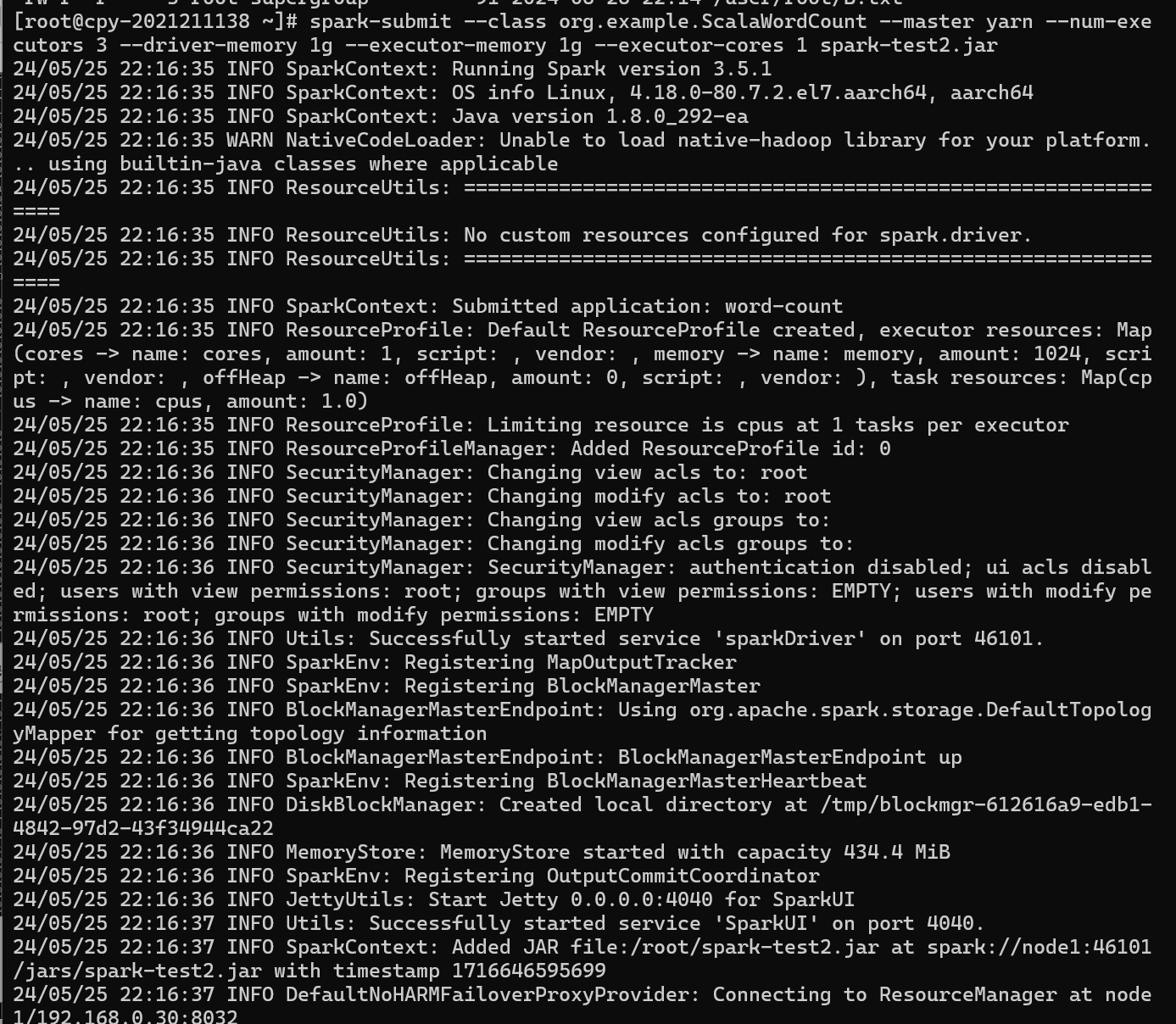
将文件上传到服务器后，移动到hdfs文件系统下：

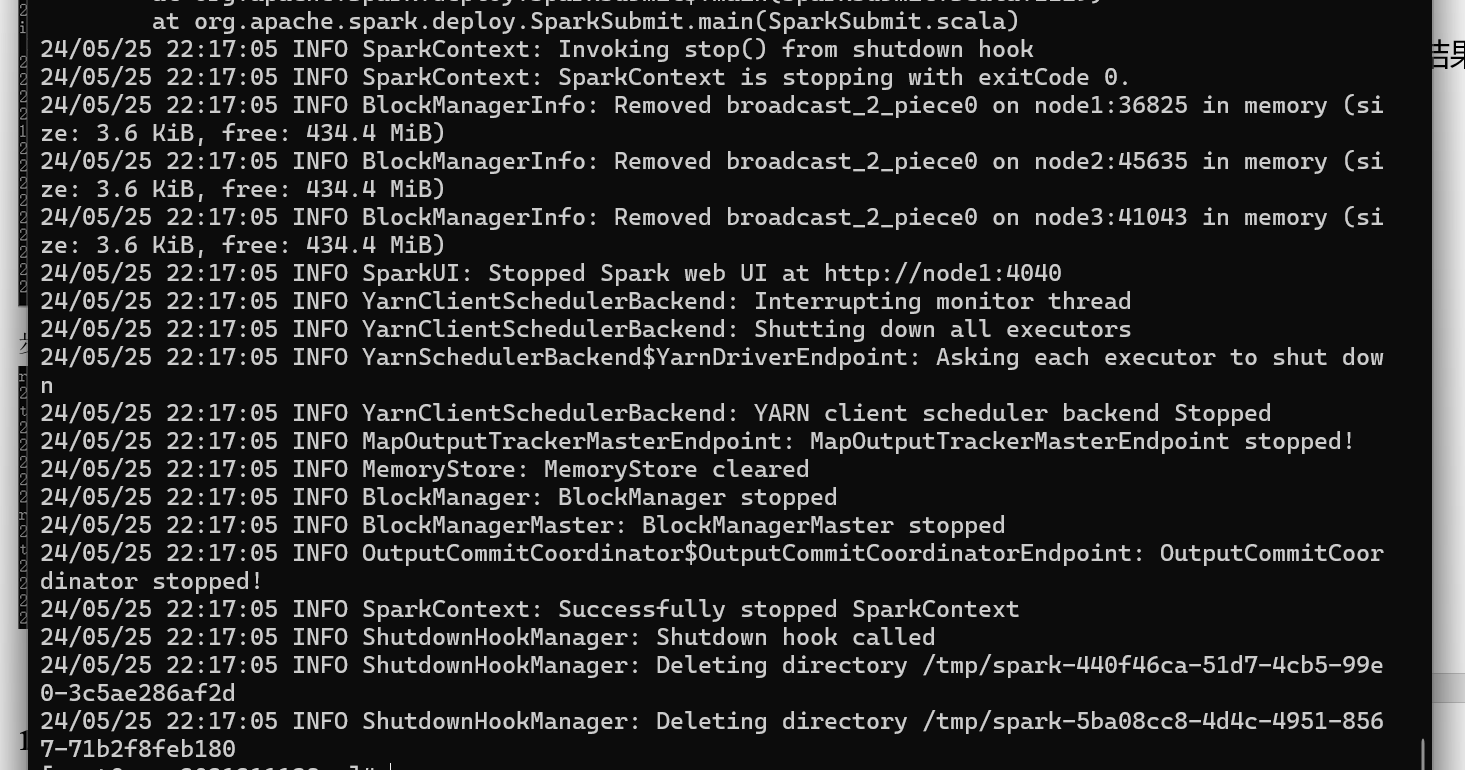


输入如下命令运行：

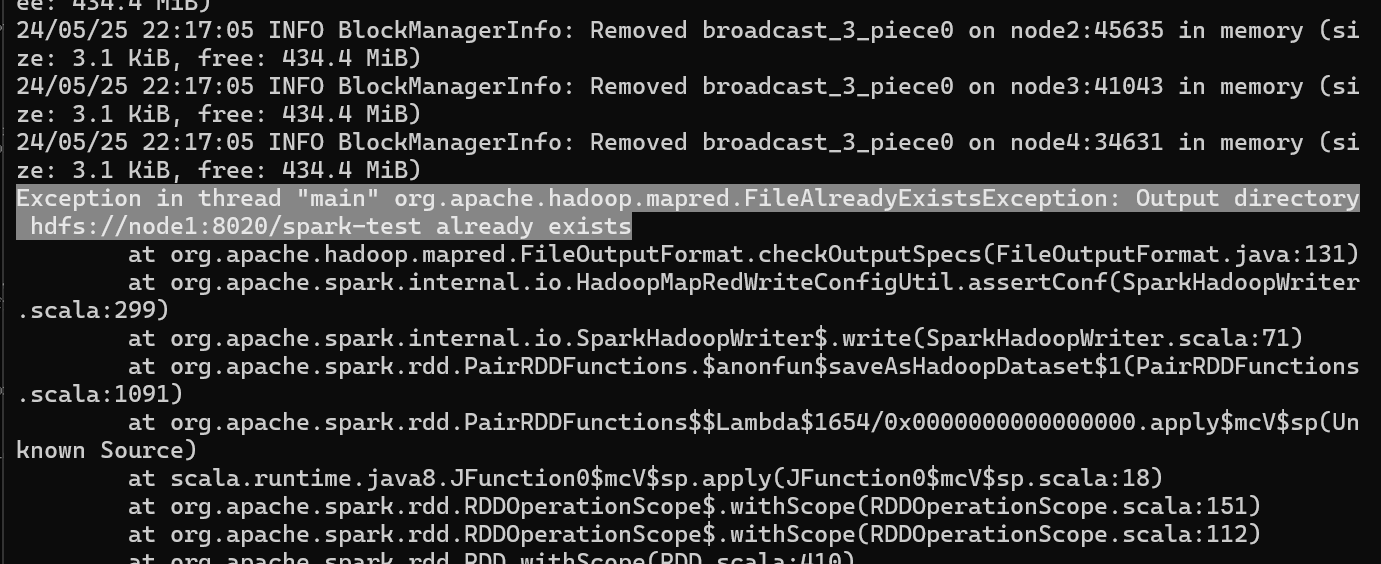
spark-submit --class org.example.ScalaWordCount --master yarn --num-executors 3 --driver-memory 1g --executor-memory 1g --executor-cores 1 spark-test2.jar

得到如下结果:





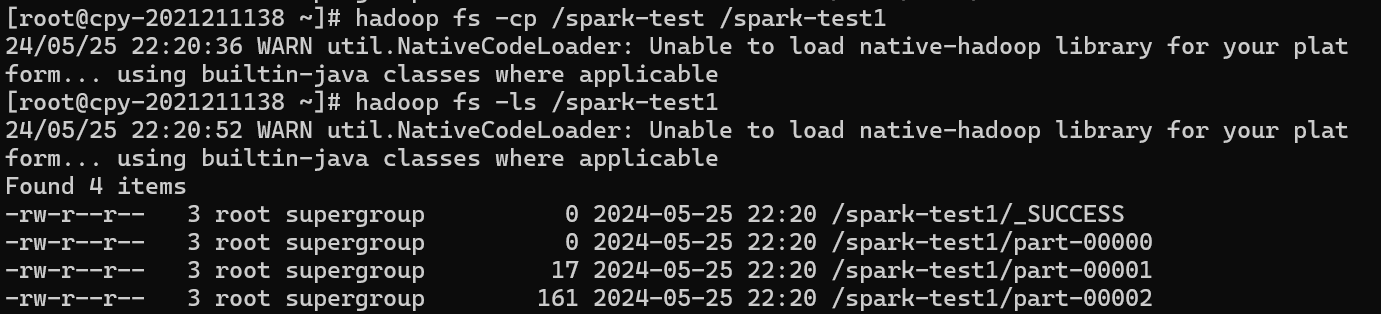
但是出现报错：



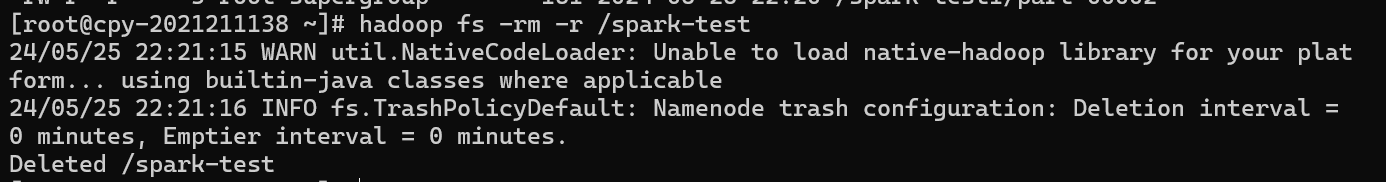
它说已经存在该路径了

那么就将原来test1的输出文件夹改为spark-test1，然后再将spark-test文件夹删掉，步骤如下：

复制，查看

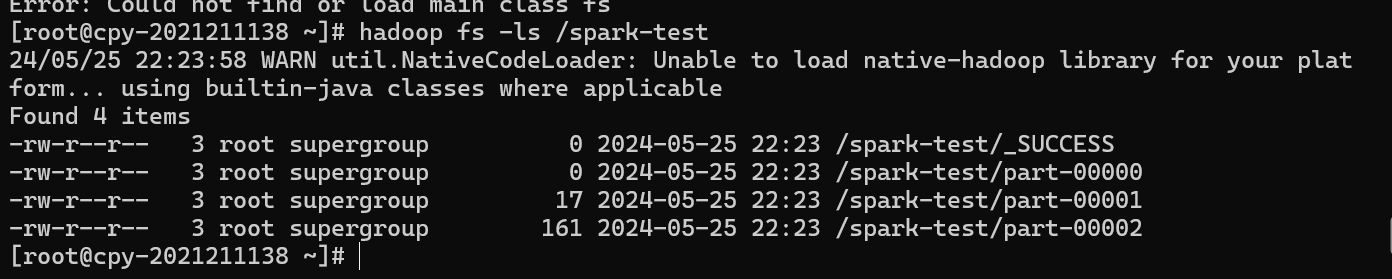


删除

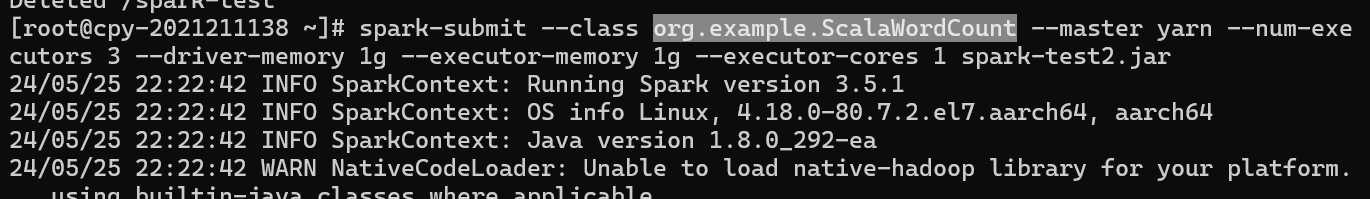


重新运行：

发现结果错误

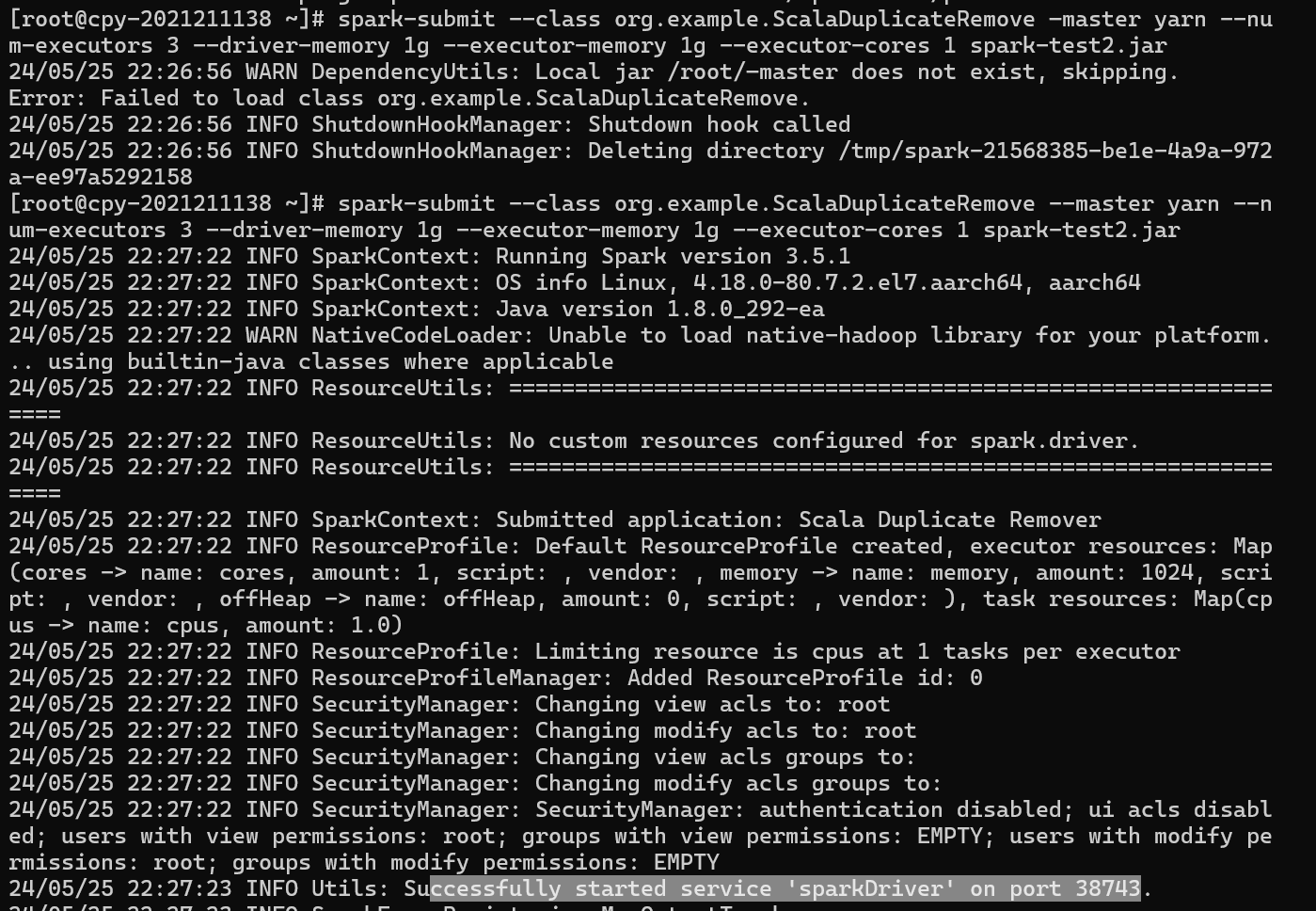


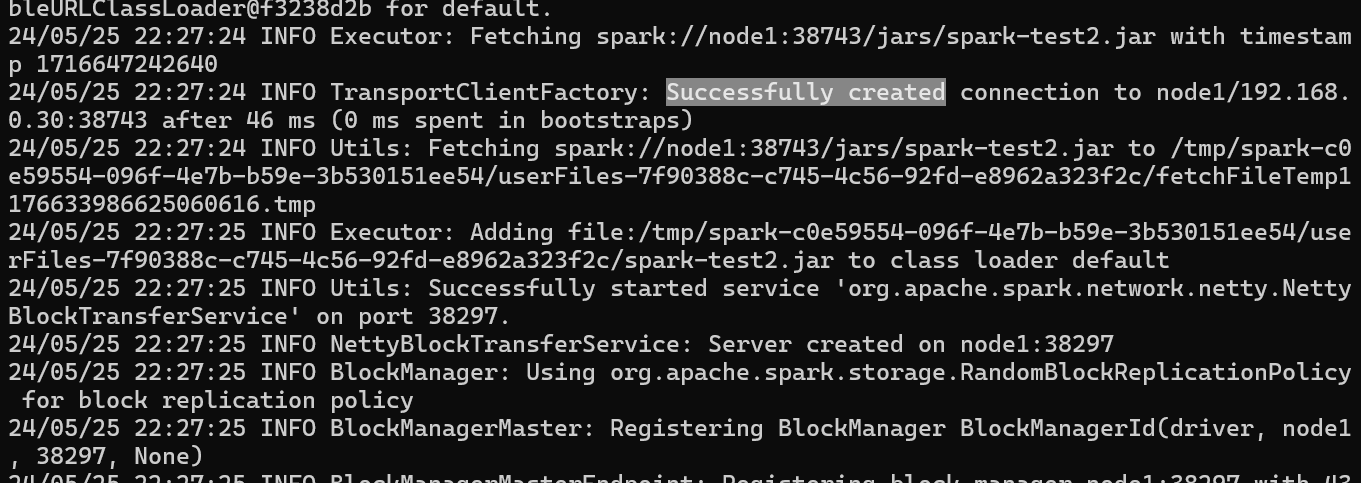
问题在于，运行的命令输错了



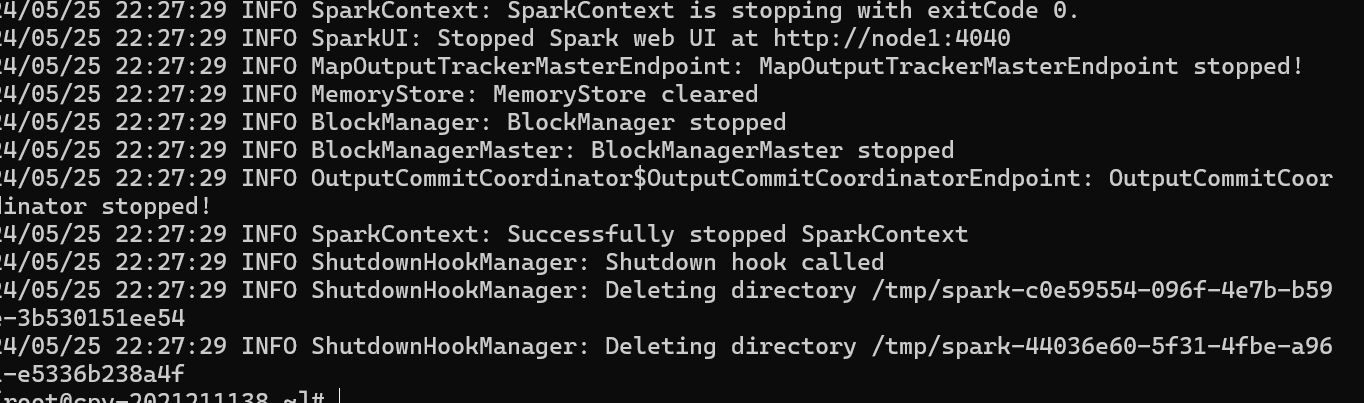
不是这个类，而是另一个，重新输入指令，如下：

spark-submit --class org.example.ScalaDuplicateRemove --master yarn --num-executors 3 --driver-memory 1g --executor-memory 1g --executor-cores 1 spark-test2.jar



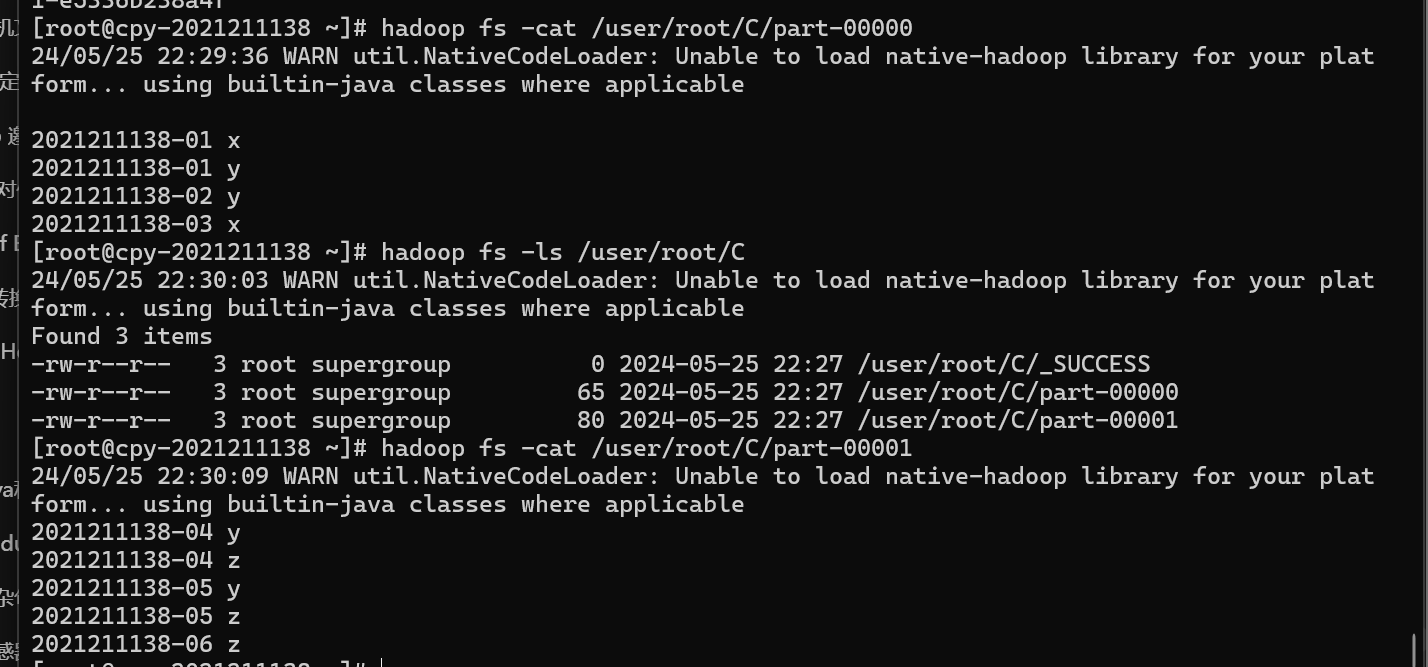






运行命令查看执行结果：

hadoop fs -cat /user/root/C/part-00000



进行实验3：

在master节点中下载并安装MySQL

使用sudo yum update 更新

使用wget命令下载mysql

wget https://dev.mysql.com/get/Downloads/MySQL-8.0/mysql-8.0.37-1.sl15.x86\_64.rpm-bundle.tar



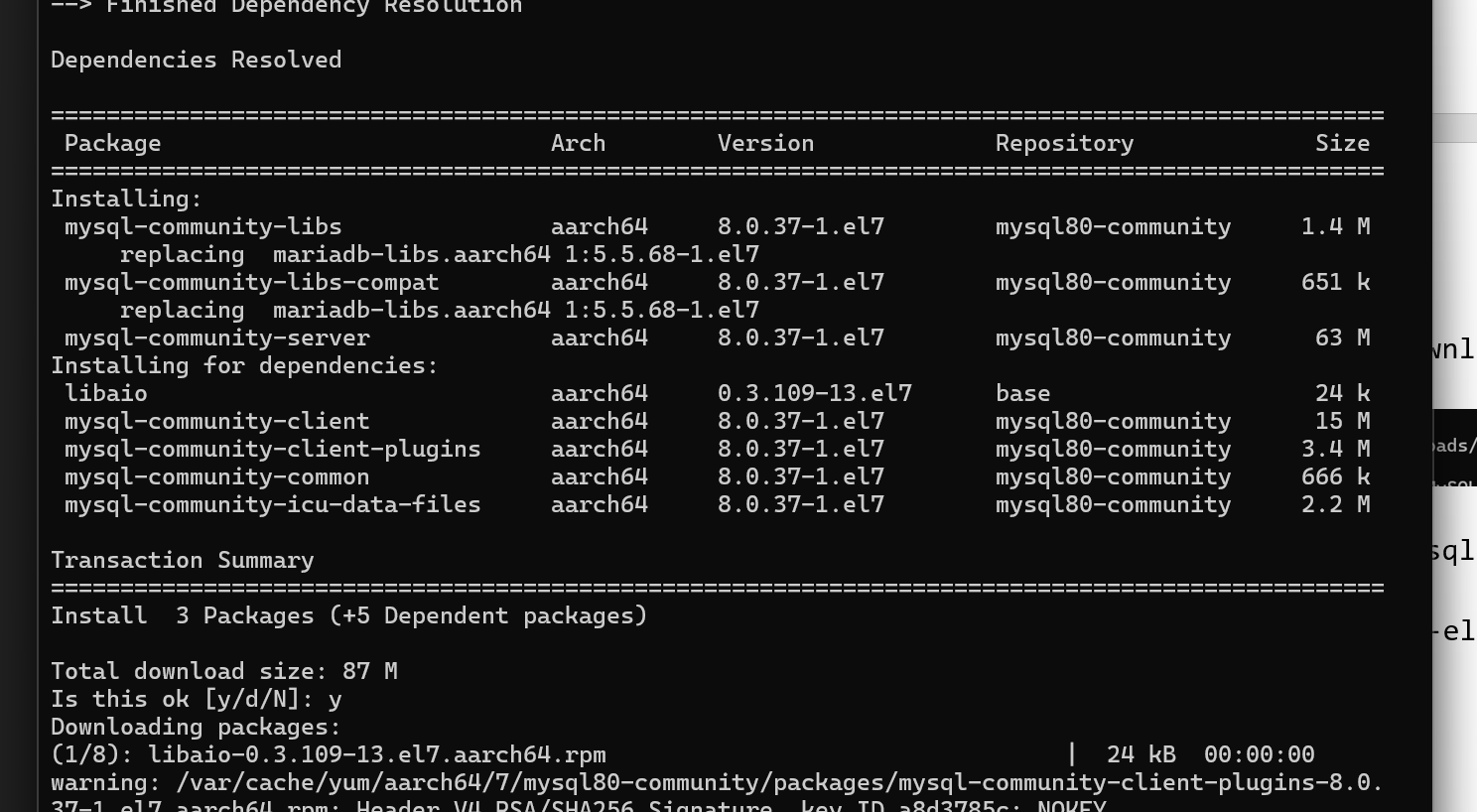
sudo wget https://dev.mysql.com/get/mysql80-community-release-el7-3.noarch.rpm

sudo rpm -ivh mysql80-community-release-el7-3.noarch.rpm

sudo yum update

sudo yum install mysql-server

之后运行上面四个命令



就会开始下载mysql-server

安装mysql出现报错：

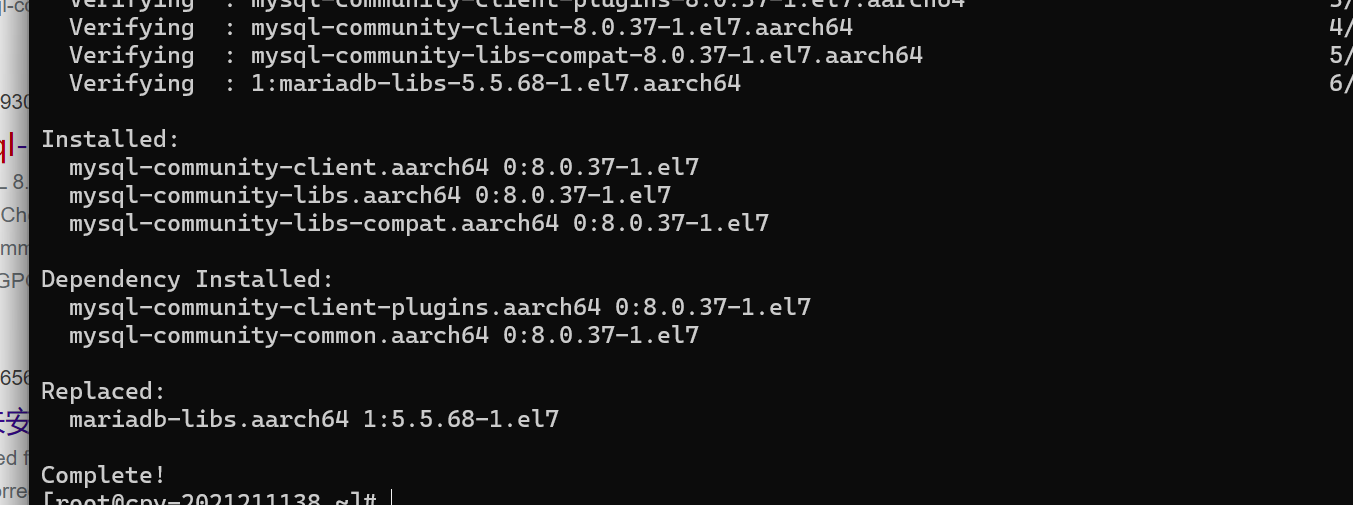
The GPG keys listed for the "MySQL 8.0 Community Server" repository are already installed but they are not correct for this package. Check that the correct key URLs are configured for this repository. Failing package is: mysql-community-common-8.0.37-1.el7.aarch64 GPG Keys are configured as: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

这是因为GPG公钥验证问题

输入：

sudo rpm --import <https://repo.mysql.com/RPM-GPG-KEY-mysql-2023>

就能解决



下载完成

安装mysql-devel

yum install mysql-devel

安装mysql-server

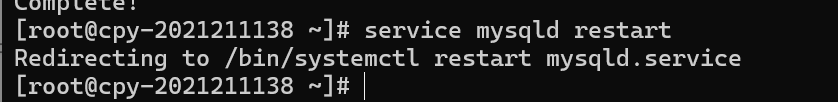
wget http://dev.mysql.com/get/mysql80-community-release-el7-5.noarch.rpm

rpm -ivh mysql80-community-release-el7-5.noarch.rpm

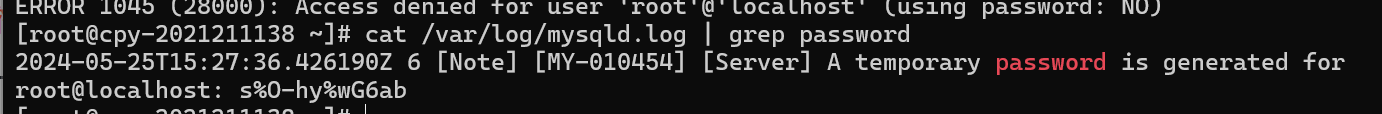
yum install mysql-community-server

安装成功后重启mysql服务：

service mysqld restart



初次安装，需要查看mysql的密码：

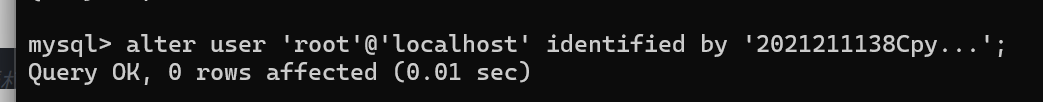


cat /var/log/mysqld.log | grep password

使用

alter user 'root'@'localhost' identified by '这里填你要的密码';

更改密码



必须有大小写和数字、字符

密码为我的学号+姓名缩写+三个点

FLUSH PRIVILEGES; 刷新权限

通过JDBC连接数据库

先输入create database spark来创建spark数据库

create database spark;

use spark;

create table student(id int(4), name char(20), gender char(4), age int(4));

insert into student values(1, 'Li', 'F', 23);

insert into student values(2, 'Wang', 'M', 24);

select \* from student;



下载驱动程序：

wget <https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-8.0.24.tar.gz>

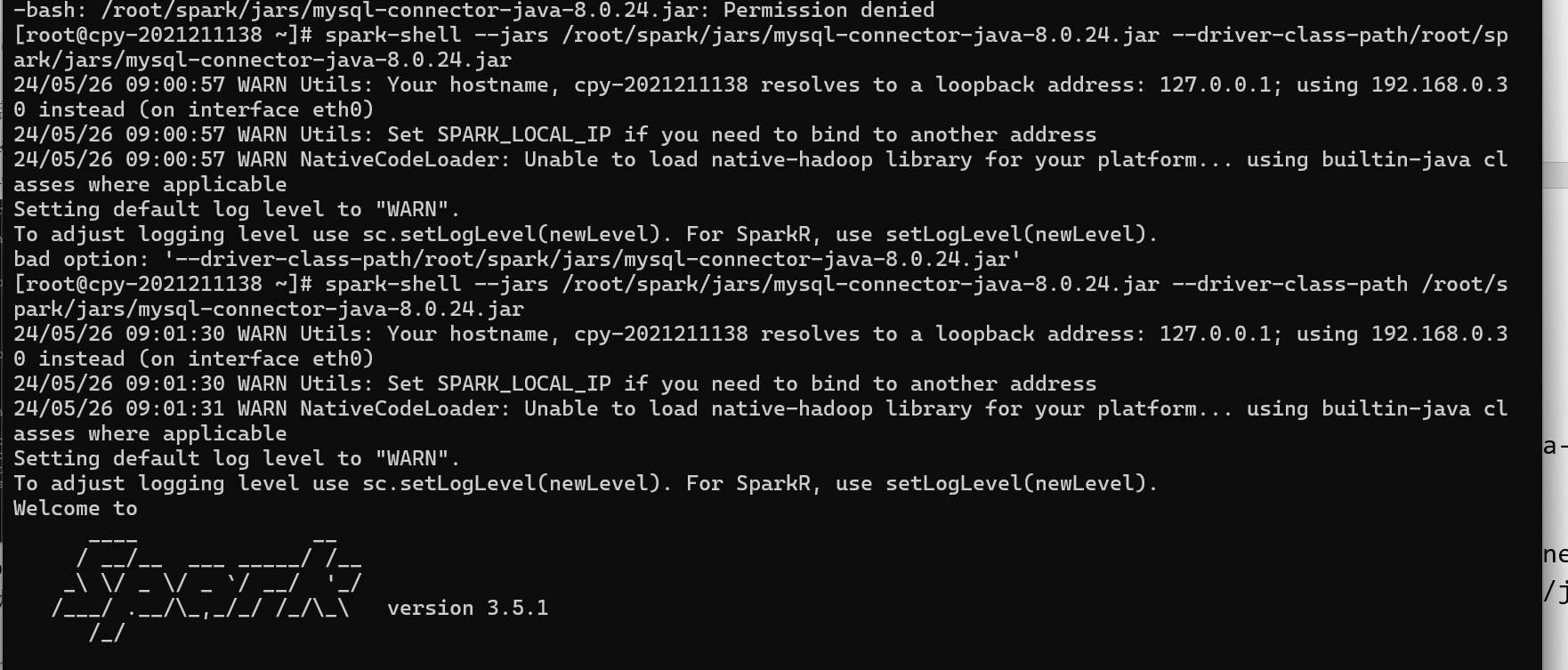
解压：

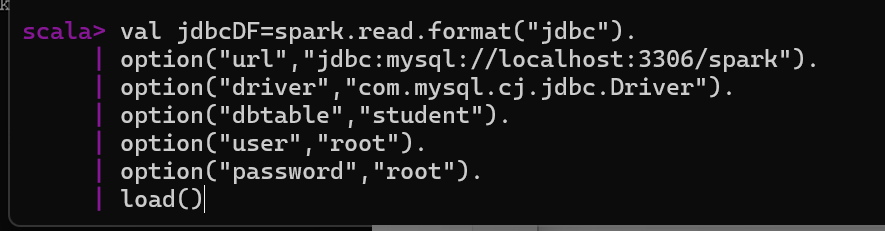
tar -xzvf mysql-connector-java-8.0.24.tar.gz

mv mysql-connector-java-8.0.24/mysql-connector-java-8.0.24.jar /root/spark/jars/

启动spark shell时，必须指定MySQL连接驱动jar包，命令如下

spark-shell --jars /root/spark/jars/mysql-connector-java-8.0.24.jar --driver-class-path /root/spark/jars/mysql-connector-java-8.0.24.jar





val jdbcDF=spark.read.format("jdbc").

option("url","jdbc:mysql://localhost:3306/spark").

option("driver","com.mysql.cj.jdbc.Driver").

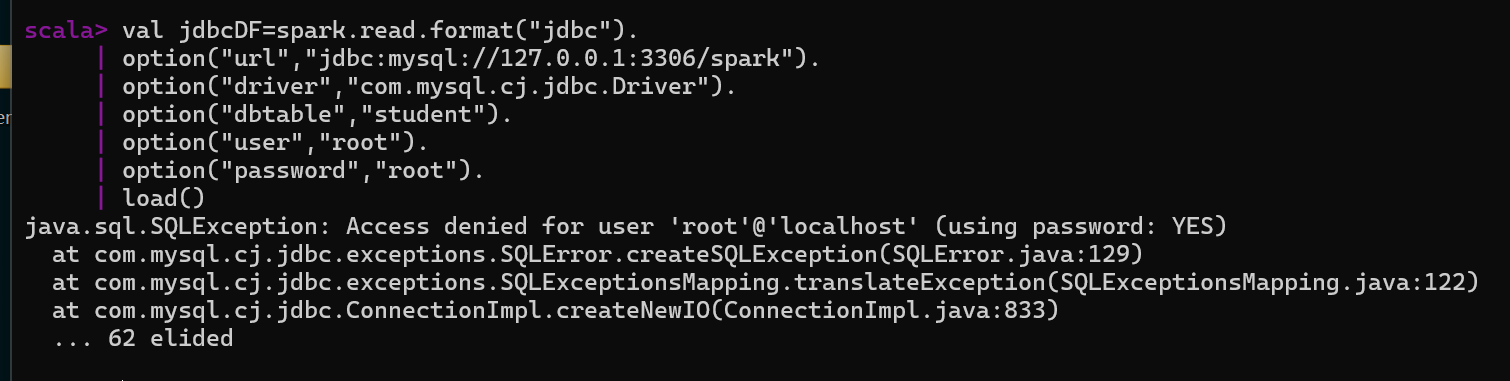
option("dbtable","student").

option("user","root").

option("password","root").

load()

出现：



原因可能是因为mysql还没启动

输入：

sudo systemctl start mysqld

启动mysql，并重新登录spark-shell

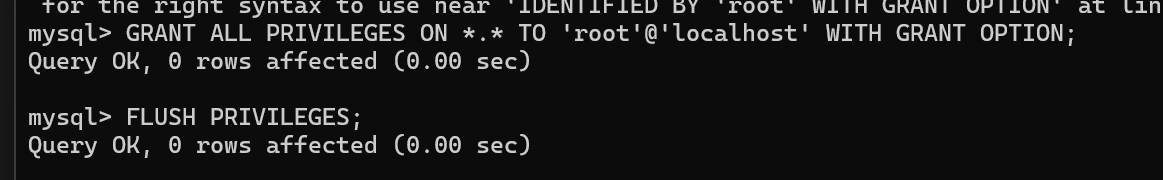
还是同样的报错。

仔细看报错原因，是access denied for user root

可能是因为root用户权限不够，要为root用户打开更高的权限

GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'localhost' WITH GRANT OPTION;

输入之后刷新



后面发现是密码输错了

我的密码和助教的不一样

val jdbcDF=spark.read.format("jdbc").

option("url","jdbc:mysql://localhost:3306/spark").

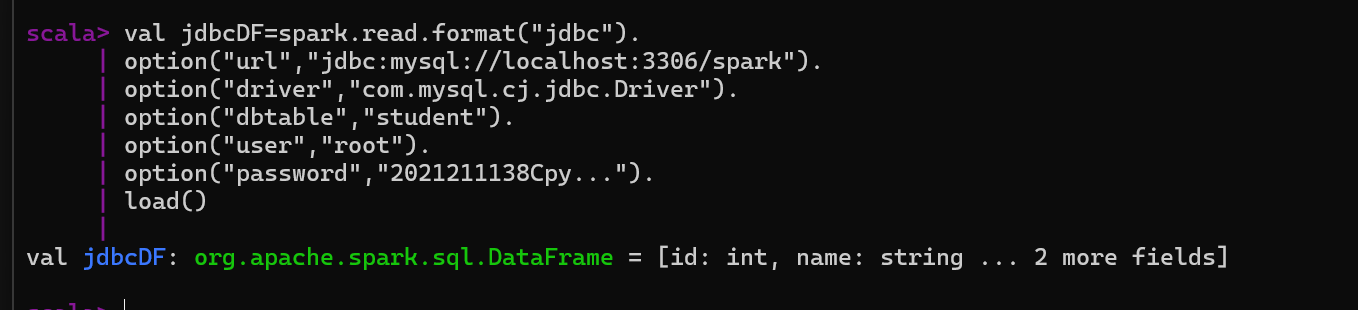
option("driver","com.mysql.cj.jdbc.Driver").

option("dbtable","student").

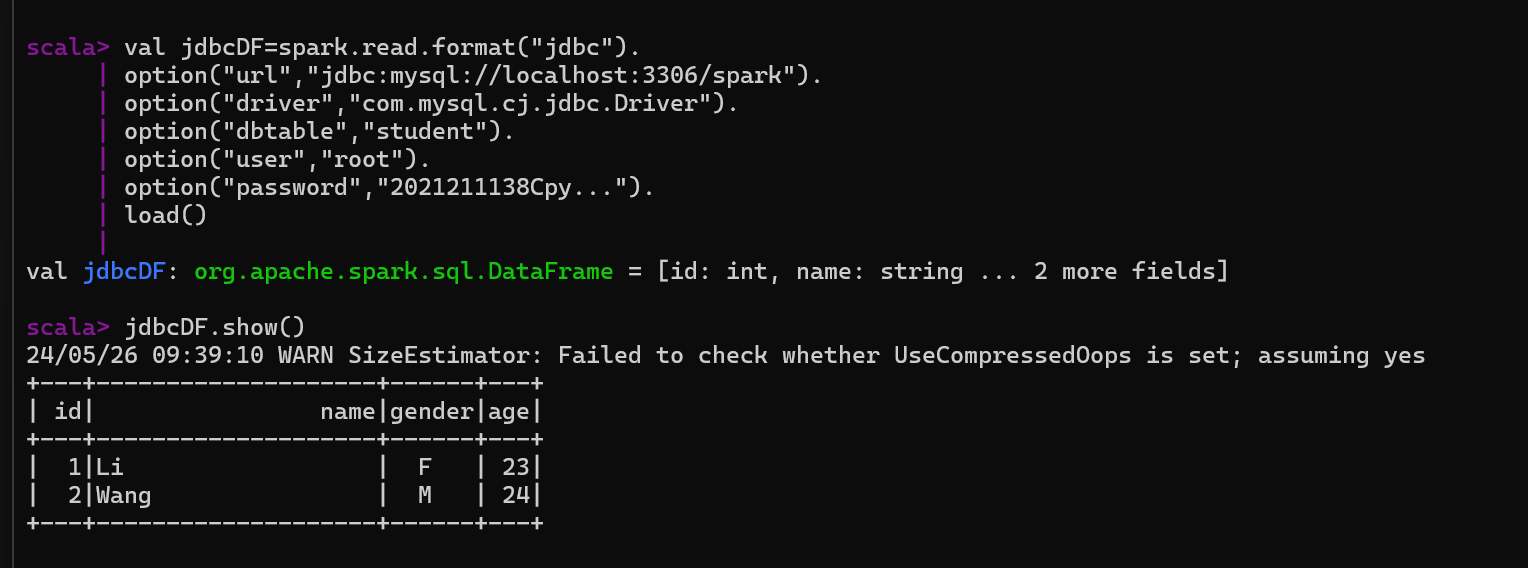
option("user","root").

option("password","2021211138Cpy...").

load()



查看表单：



向mysql数据库写入数据

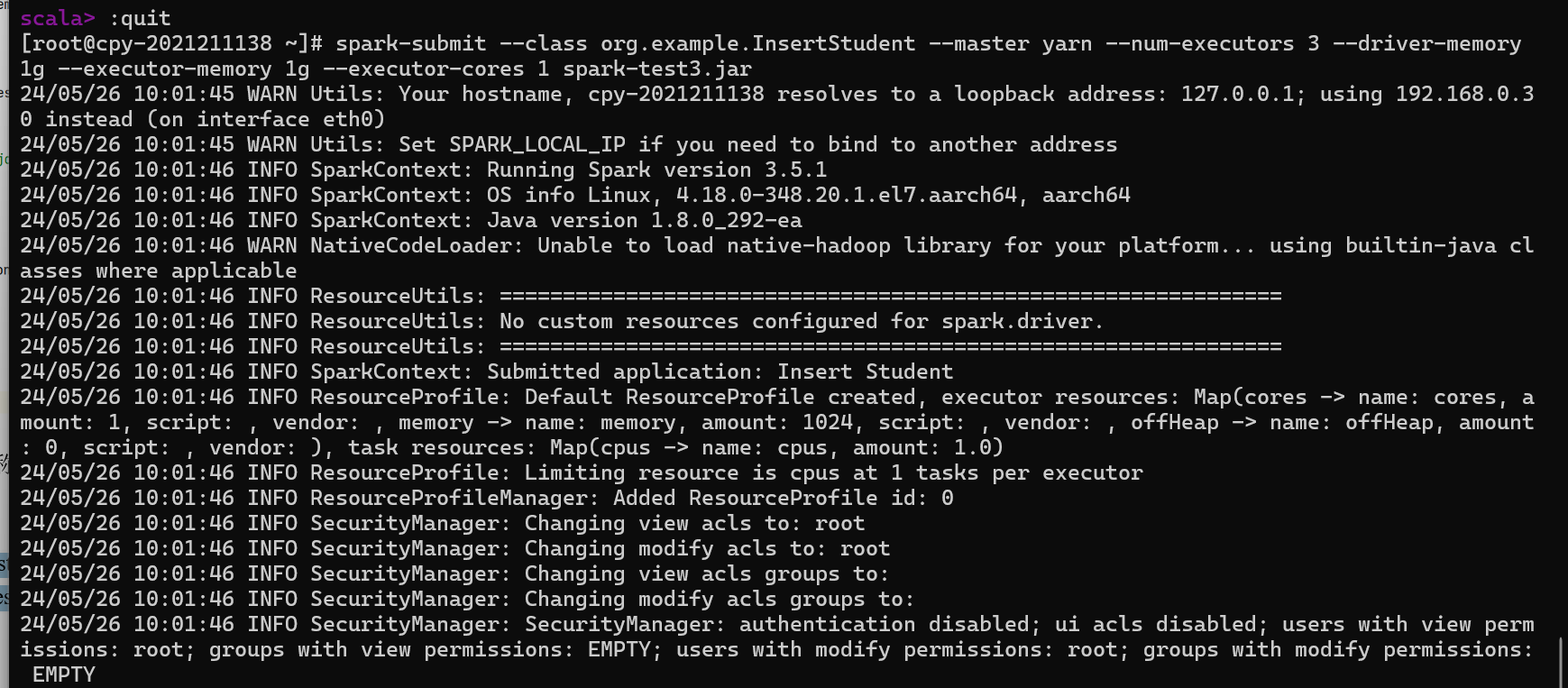
在idea中编写程序

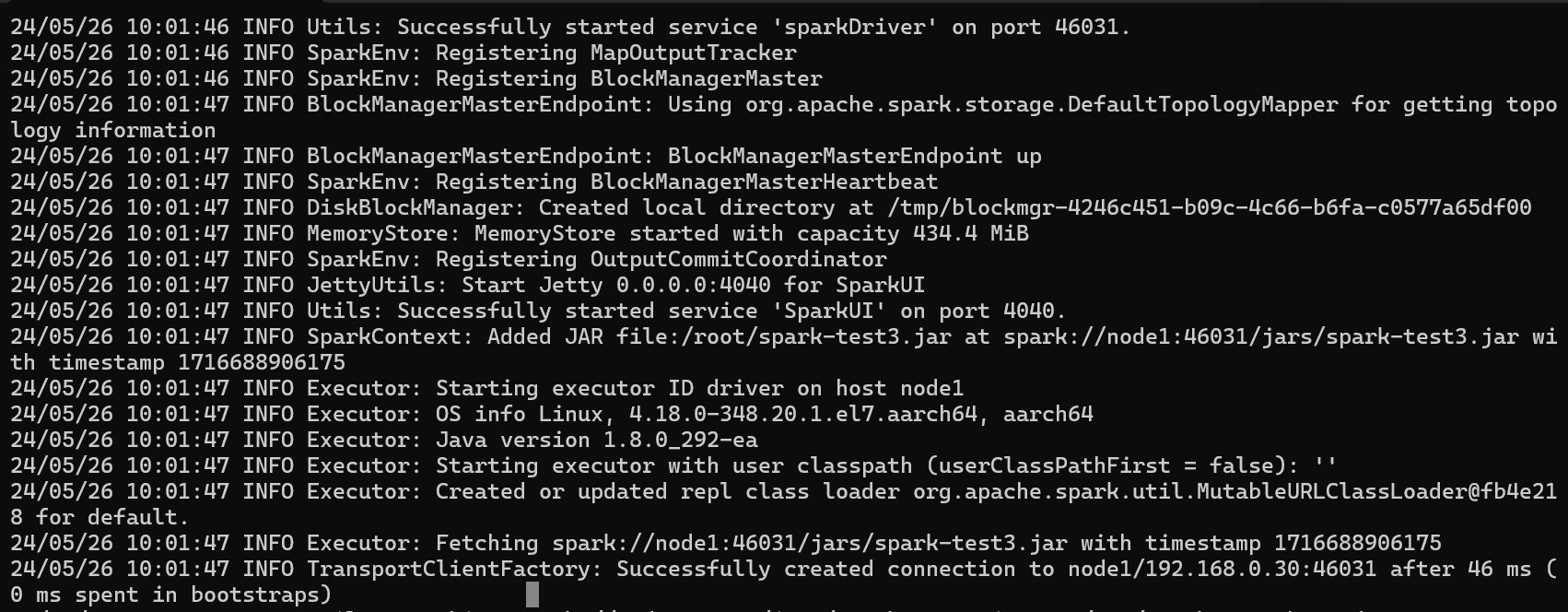
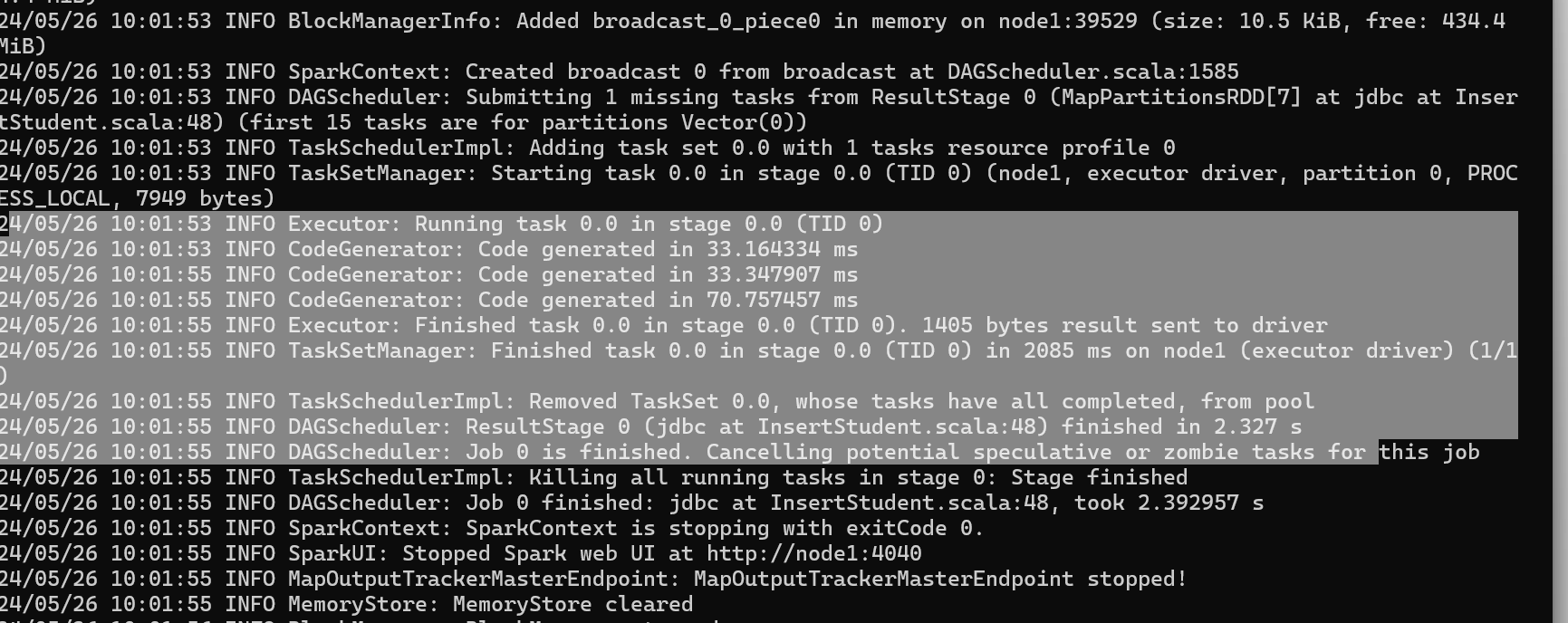
package org.example  
  
import org.apache.spark.sql.{Row, SparkSession}  
import org.apache.spark.sql.types.{IntegerType, StringType, StructField, StructType}  
  
import java.util.Properties  
  
*/\*\*  
 \** ***@author*** *陈朴炎  
 \** ***@version*** *1.0  
 \*/*class InsertStudent {  
  
}  
object InsertStudent{  
 def main(args: Array[String]):Unit={  
 // 创建sparkSession  
 val spark = SparkSession.*builder*()  
 .appName("Insert Student")  
 .master("local")  
 .getOrCreate()  
 // 程序入口  
 val sc = spark.sparkContext  
 // 学生信息  
 val studentData = *Array*("3 Chen M 21","4 liu M 27")  
 val studentRDD = sc.parallelize(studentData).map(\_.split("\\s+"))  
 // 模式信息  
 val schema = *StructType*(List(  
 StructField("id",IntegerType, true),  
 StructField("name",StringType, true),  
 StructField("gender",StringType, true),  
 StructField("age",IntegerType, true)  
 ))  
 // Row对象  
 val rowRDD = studentRDD.map(attrs => *Row*(attrs(0).toInt, attrs(1), attrs(2), attrs(3).toInt))  
 // 建立DataFrame  
 val studentDF = spark.createDataFrame(rowRDD,schema)  
 // 设置连接的参数  
 val jdbcUrl = "jdbc:mysql://localhost:3306/spark"  
 val connectionProperties = new Properties()  
 connectionProperties.put("user","root")  
 connectionProperties.put("password","2021211138Cpy...")  
 connectionProperties.put("driver","com.mysql.cj.jdbc.Driver")  
  
 // 将数据写入数据库  
 studentDF.write  
 .mode("append")  
 .jdbc(jdbcUrl, "spark.student", connectionProperties)  
  
 // 释放  
 spark.stop()  
 }  
}

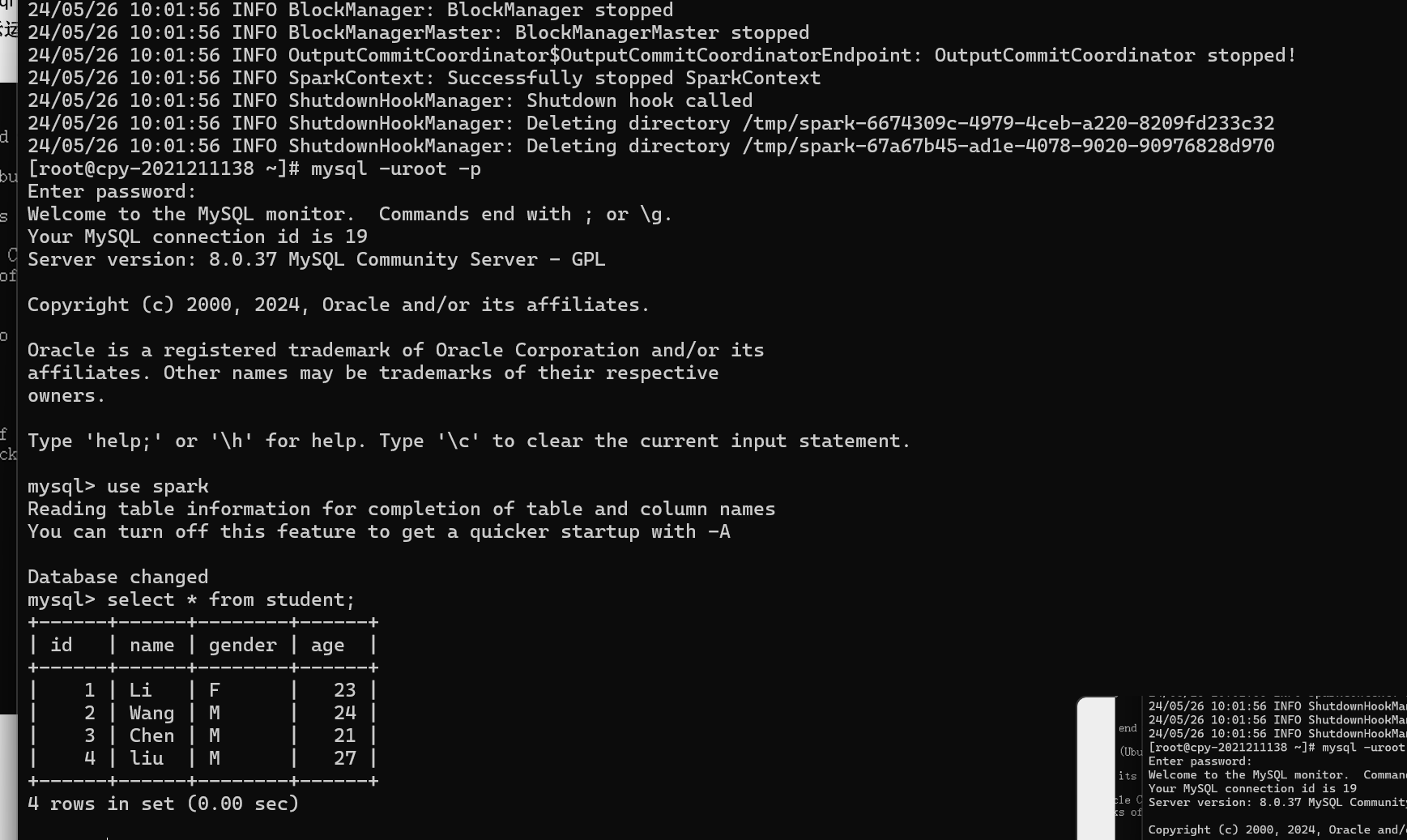
同样的，打包，然后删除MANIFEST.MF，上传

运行：

spark-submit --class org.example.InsertStudent --master yarn --num-executors 3 --driver-memory 1g --executor-memory 1g --executor-cores 1 spark-test3.jar





成功。