Windows配置Hadoop

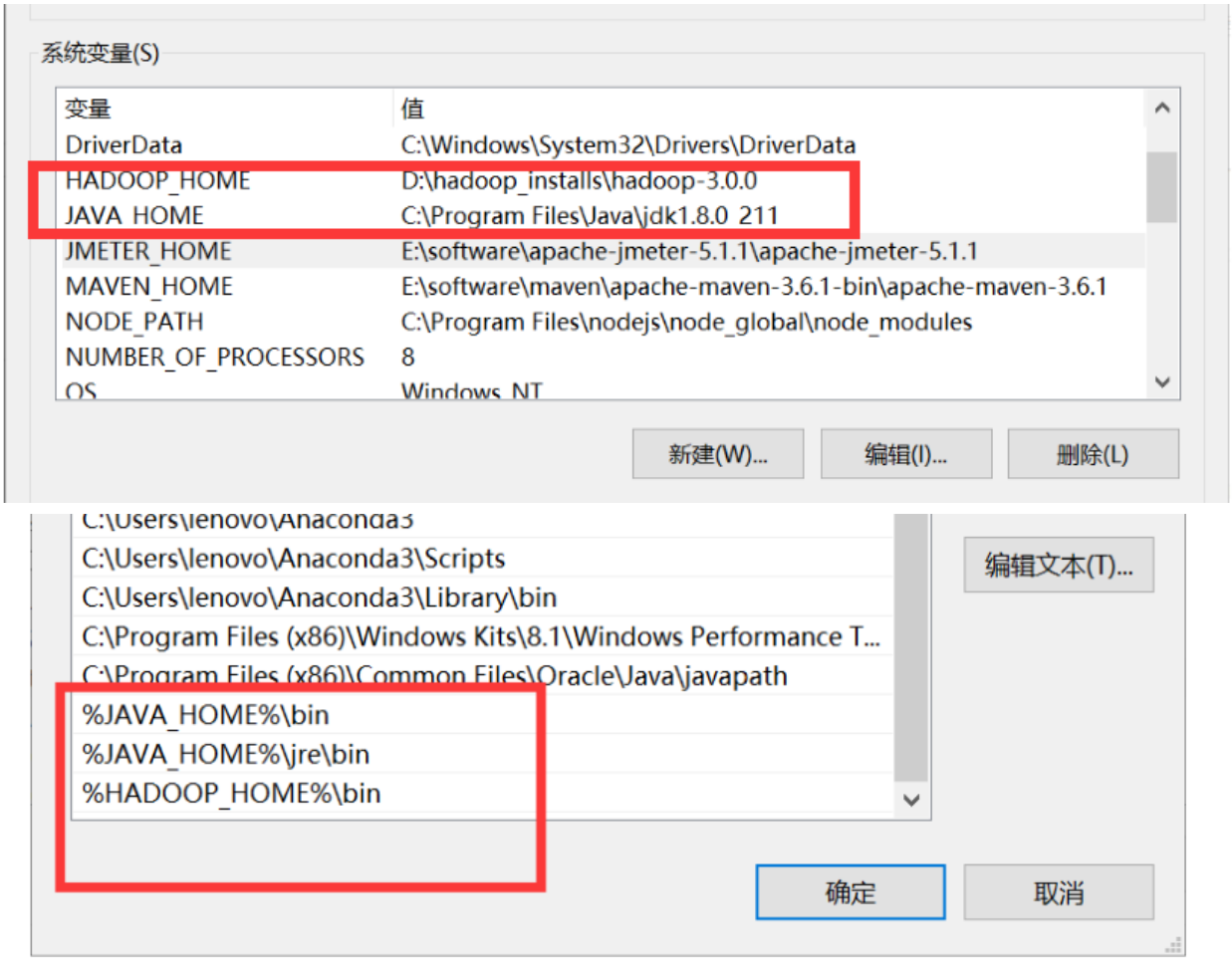
目录

Windows下Hadoop-3.0.0安装运行

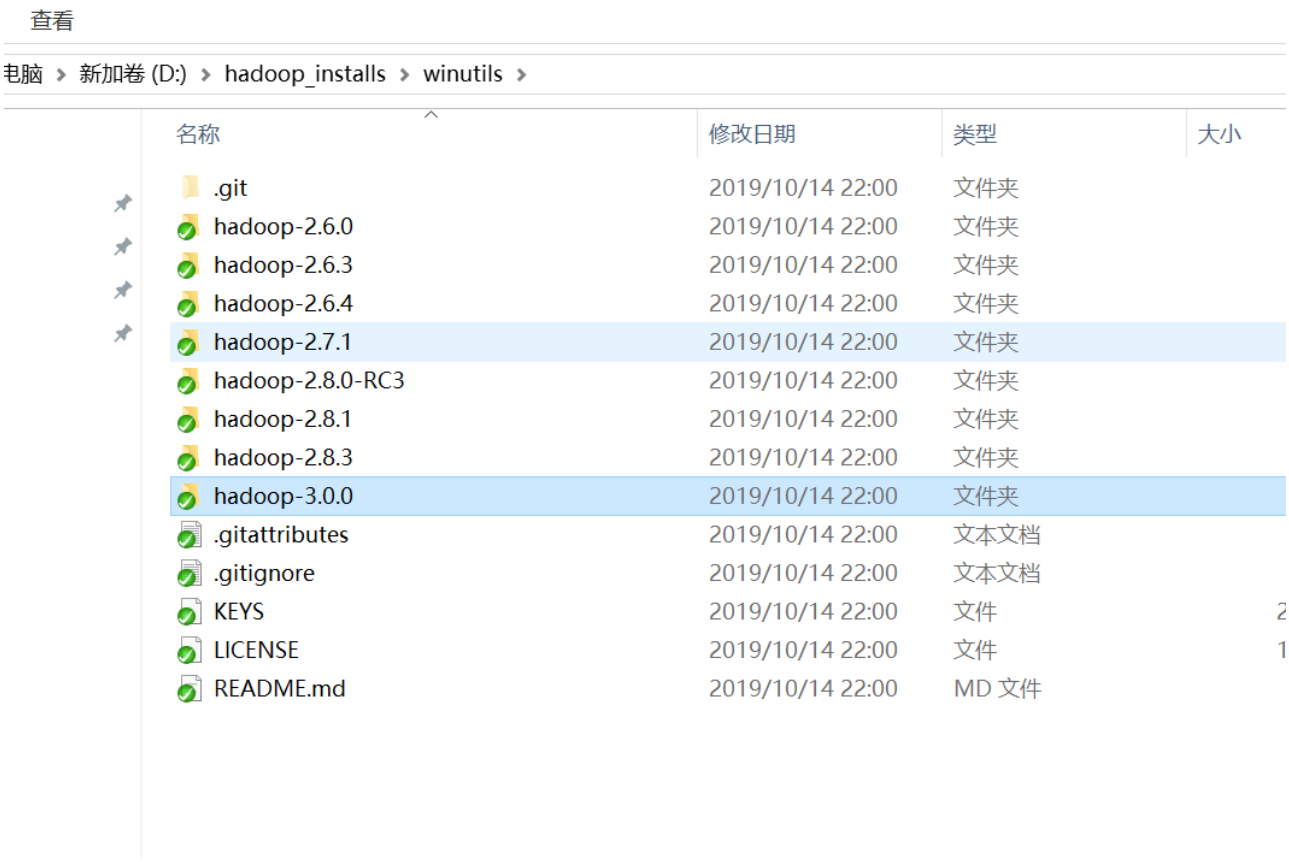
1. 建立hadoop_installs文件夹，官网下载 hadoop-3.0.0 binary到该目录
2. 管理员权限打开shell，解压后出现hadoop-3.0.0目录

```
tar -zxvf hadoop-3.0.0.tar.gz
```

3. 环境变量配置



4. 添加winutils https://github.com/steveloughran/winutils clone到本地解压对应版本的替换bin目录（或者将winutils.exe直接放bin里）



5. 修改hadoop配置文件

   - hadoop-3.0.0/etc/hadoop/core-site.xml配置

```
<configuration>
    <property>
       <name>fs.default.name</name>
       <value>hdfs://localhost:9000</value>
   </property>
</configuration>
```

   - hadoop-3.0.0/etc/hadoop/mapred-site.xml配置

```
<configuration>
     <property>
         <name>mapreduce.framework.name</name>
         <value>yarn</value>
     </property>
</configuration>
```

   - hadoop-3.0.0目录下创建tmp目录，hadoop-3.0.0/tmp下再创建datanode目录和namenode目录
   - hadoop-3.0.0/etc/hadoop/hdfs-site.xml配置

```
 <configuration>
     <!-- 这个参数设置为1，因为是单机版hadoop -->
     <property>
         <name>dfs.replication</name>
         <value>1</value>
     </property>
     <property>
         <name>dfs.permissions</name>
         <value>false</value>
     </property>
     <property>
         <name>dfs.namenode.name.dir</name>
         <value>/D:/hadoop_installs/hadoop-3.0.0/tmp/namenode</value>
     </property>
     <property>
             <name>fs.checkpoint.dir</name>
             <value>/D:/hadoop_installs/hadoop-3.0.0/tmp/snn</value>
         </property>
         <property>
             <name>fs.checkpoint.edits.dir</name>
             <value>/D:/hadoop_installs/hadoop-3.0.0/tmp/snn</value>
         </property>
         <property>
         <name>dfs.datanode.data.dir</name>
         <value>/D:/hadoop_installs/hadoop-3.0.0/tmp/datanode</value>
     </property>
 </configuration>
```

- hadoop-3.0.0/etc/hadoop/yarn-site.xml配置

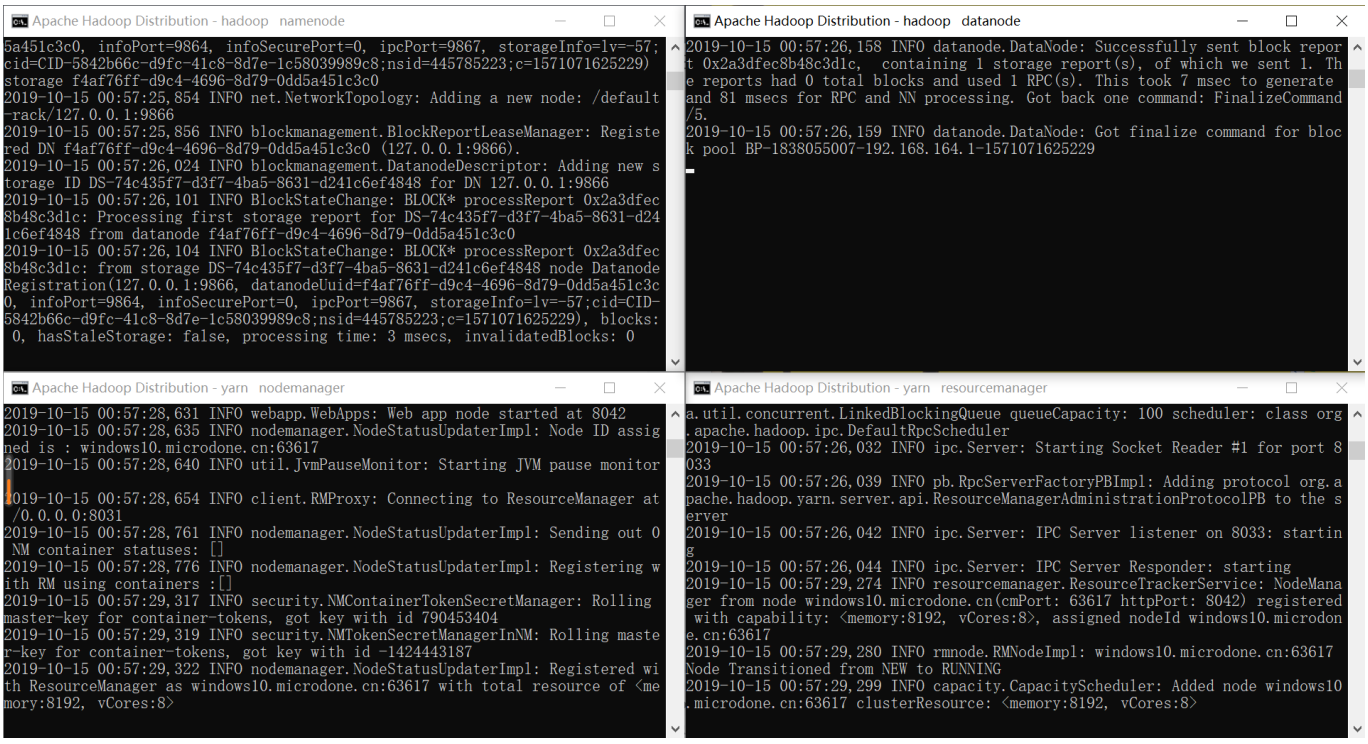
```
<configuration>
<!-- Site specific YARN configuration properties -->
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>
    <property>

<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
            <value>org.apache.hadoop.mapred.ShuffleHandler</value>
    </property>
</configuration>
```

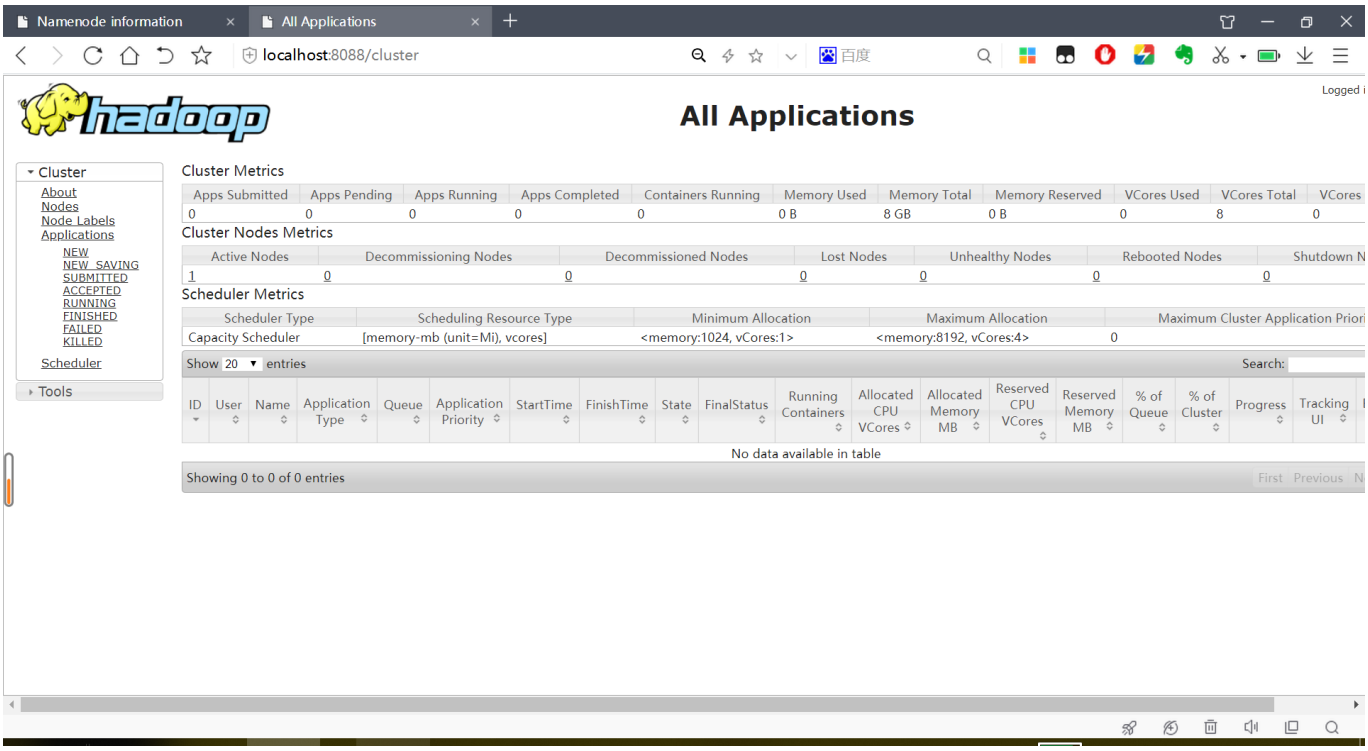
6. hadoop-3.0.0/etc/hadoop/**hadoop-env.cmd**配置

set JAVA_HOME=%JAVA_HOME%"替换为"set JAVA_HOME=set JAVA_HOME=C:\PROGRA~1\Java\jdk1.8.0_211"，路径不能有空格否则会报错,program files用PROGRA~1替换

7. 启动服务

```
bin\hdfs namenode-format
sbin\start-all.cmd
```



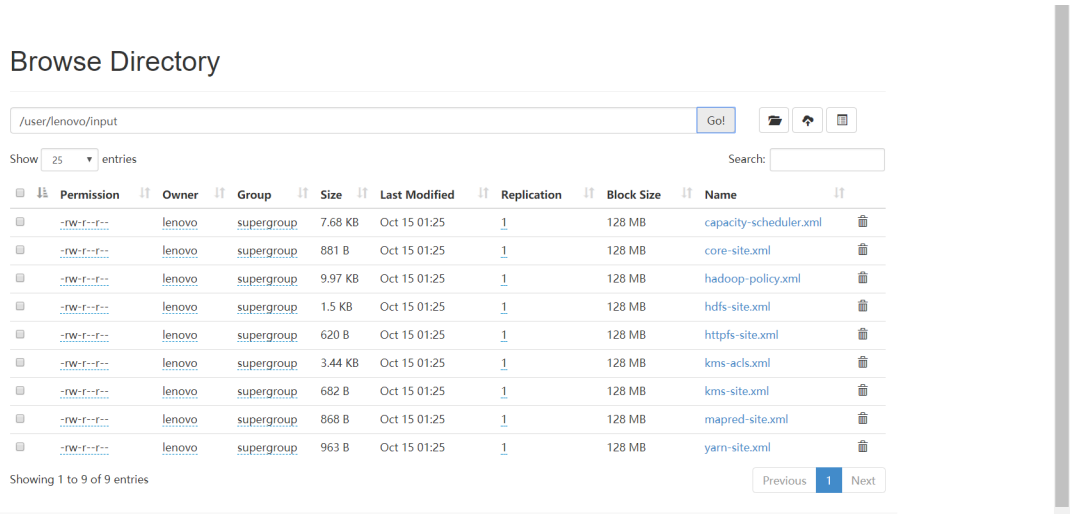


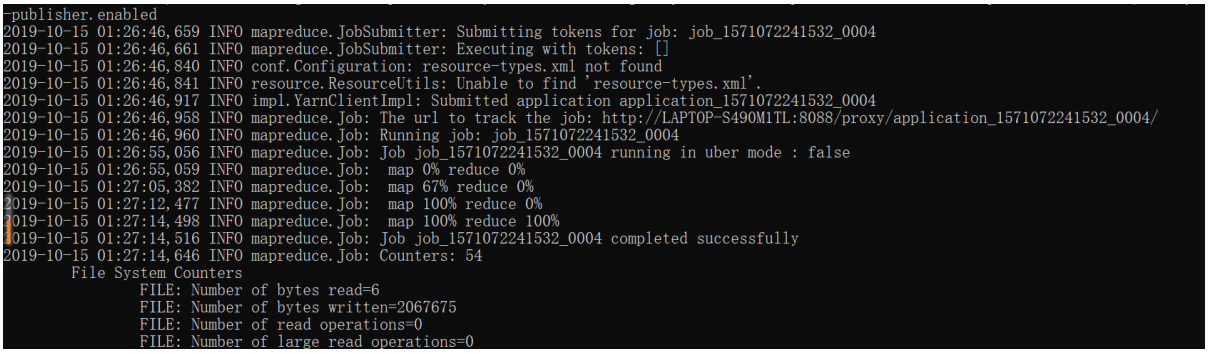
8. 运行grep示例
   - 操作hdfs和linux下一样，不再赘述，似乎要用绝对路径...
   - grep卡在map 0% reduce 100%，报错内容：path is not a file，解决方案：hdfs的input改成只放进去文件/*.xml而不是文件夹，成功运行

```
        at org.apache.hadoop.io.retry.RetryInvocationHandler$Call.invoke(RetryInvocationHandler.java:157)
        at org.apache.hadoop.io.retry.RetryInvocationHandler$Call.invokeOnce(RetryInvocationHandler.java:
        at org.apache.hadoop.io.retry.RetryInvocationHandler.invoke(RetryInvocationHandler.java:359)
        at com.sun.proxy.$Proxy18.getBlockLocations(Unknown Source)
        at org.apache.hadoop.hdfs.DFSClient.callGetBlockLocations(DFSClient.java:858)
        ... 17 more

2019-10-15 01:21:35,428 INFO mapreduce.Job:  map 100% reduce 100%
2019-10-15 01:21:35,448 INFO mapreduce.Job: Job job_1571072241532_0002 failed with state FAILED due to: T
Job failed as tasks failed. failedMaps:1 failedReduces:0 killedMaps:0 killedReduces: 0

2019-10-15 01:21:35,663 INFO mapreduce.Job: Counters: 9
        Job Counters
                Failed map tasks=4
                Killed reduce tasks=1
                Launched map tasks=4
                Other local map tasks=4
                Total time spent by all maps in occupied slots (ms)=12951
                Total time spent by all reduces in occupied slots (ms)=0
```

```
        at com.sun.proxy.$Proxy18.getBlockLocations(Unknown Source)
        at org.apache.hadoop.hdfs.DFSClient.callGetBlockLocations(DFSClient.java:858)
        ... 17 more

2019-10-15 01:21:28,371 INFO mapreduce.Job: Task Id : attempt_1571072241532_0002_m_000000_2, Status : FAILED
Error: java.io.FileNotFoundException: Path is not a file: /user/lenovo/input/hadoop
        at org.apache.hadoop.hdfs.server.namenode.INodeFile.valueOf(INodeFile.java:89)
        at org.apache.hadoop.hdfs.server.namenode.INodeFile.valueOf(INodeFile.java:75)
        at org.apache.hadoop.hdfs.server.namenode.FSDirStatAndListingOp.getBlockLocations(FSDirStatAndListingOp.java:152)
        at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.getBlockLocations(FSNamesystem.java:1849)
        at org.apache.hadoop.hdfs.server.namenode.NameNodeRpcServer.getBlockLocations(NameNodeRpcServer.java:727)
        at org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolServerSideTranslatorPB.getBlockLocations(ClientNamenodeProtocolServerS
:414)
```

修改后

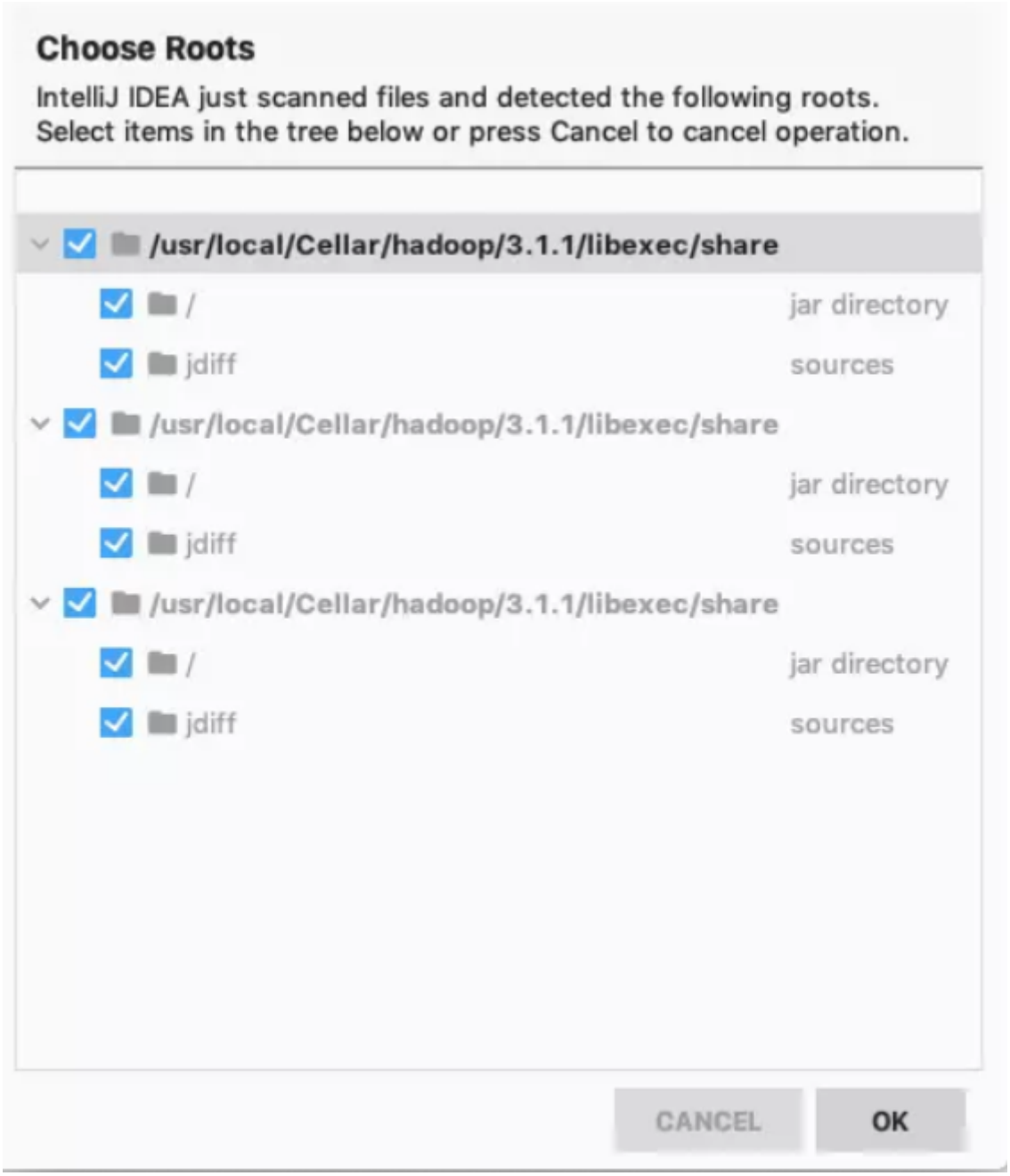


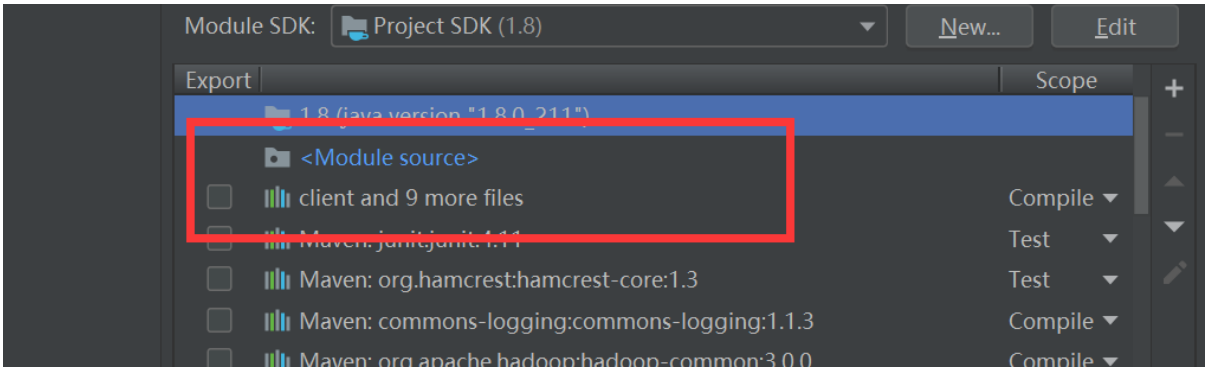


Intellij IDEA开发Hadoop伪分布式应用

坑：先关闭命令行的**java**进程再打开**IDEA**否则**IDEA**可能会打不开

1. 新建IDEA的Maven项目
    - File-New-Project-Maven，project SDK 1.8 再点击next
    - 填写groupid, artifactid，next, finish
2. 修改 Target bytecode version
    - 打开 Setting，选中 Build, Execution, Deployment -> Compiler -> java，将 Target bytecode version 改为 1.8 或 8
    - 确认一下project stucture中的Project SDK, Modules SDK都是1.8
3. 导入依赖jar包
    - 在Project Stucture中Modules选中 Dependencies 后点击下方的 + 号，选择「JARs or directories」，把hadoop目录下的share/hadoop/下的几个文件夹里的jar包都添进去





4. 在pom.xml中添加依赖 我的是这样写的

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
```

```
    <modelVersion>4.0.0</modelVersion>

    <groupId>hts.nju.edu.cn</groupId>
    <artifactId>testWordCount</artifactId>
    <version>1.0-SNAPSHOT</version>

    <dependencies>
        <!-- https://mvnrepository.com/artifact/junit/junit -->
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.11</version>
            <scope>test</scope>
        </dependency>
        <!--&lt;!&ndash; https://mvnrepository.com/artifact/commons-
logging/commons-logging &ndash;&gt;-->
        <dependency>
            <groupId>commons-logging</groupId>
            <artifactId>commons-logging</artifactId>
            <version>1.1.3</version>
        </dependency>
        <!--&lt;!&ndash;
https://mvnrepository.com/artifact/org.apache.hadoop/hadoop-common
&ndash;&gt;-->
        <dependency>
            <groupId>org.apache.hadoop</groupId>
            <artifactId>hadoop-common</artifactId>
            <version>3.0.0</version>
        </dependency>
        <dependency>
            <groupId>org.apache.hadoop</groupId>
            <artifactId>hadoop-hdfs</artifactId>
            <version>3.0.0</version>
        </dependency>
        <dependency>
            <groupId>org.apache.hadoop</groupId>
            <artifactId>hadoop-client</artifactId>
            <version>3.0.0</version>
        </dependency>
    </dependencies>
</project>
```

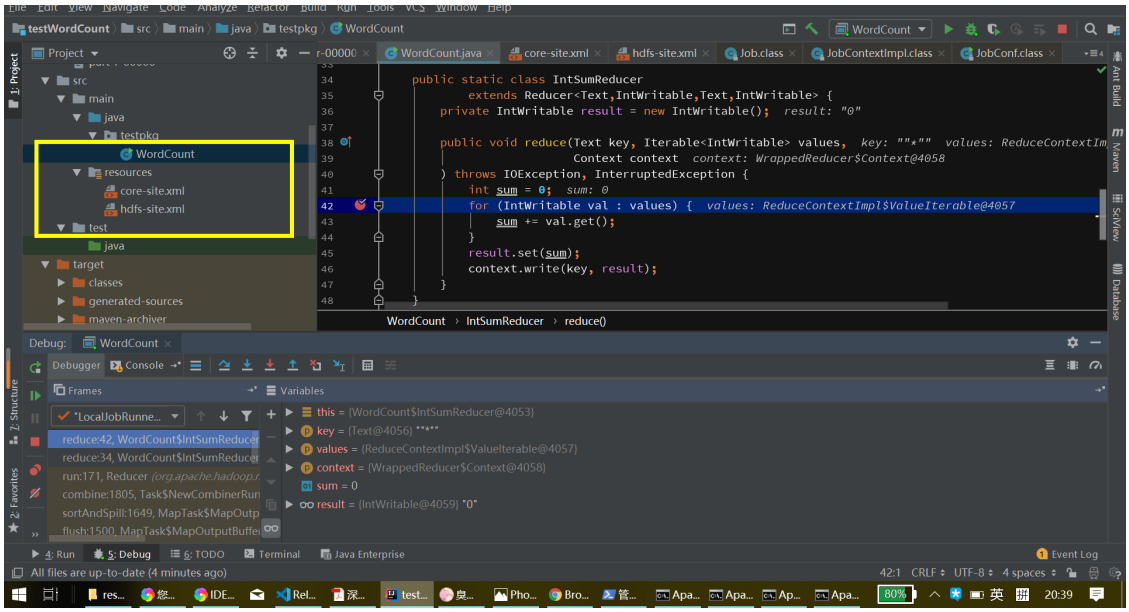
- 添加后刷新maven下载依赖，注意有的下载错误的，不方，可以去本地maven仓库删除掉再重新下载或许就好了。

5. 编写hadoop java代码WordCount

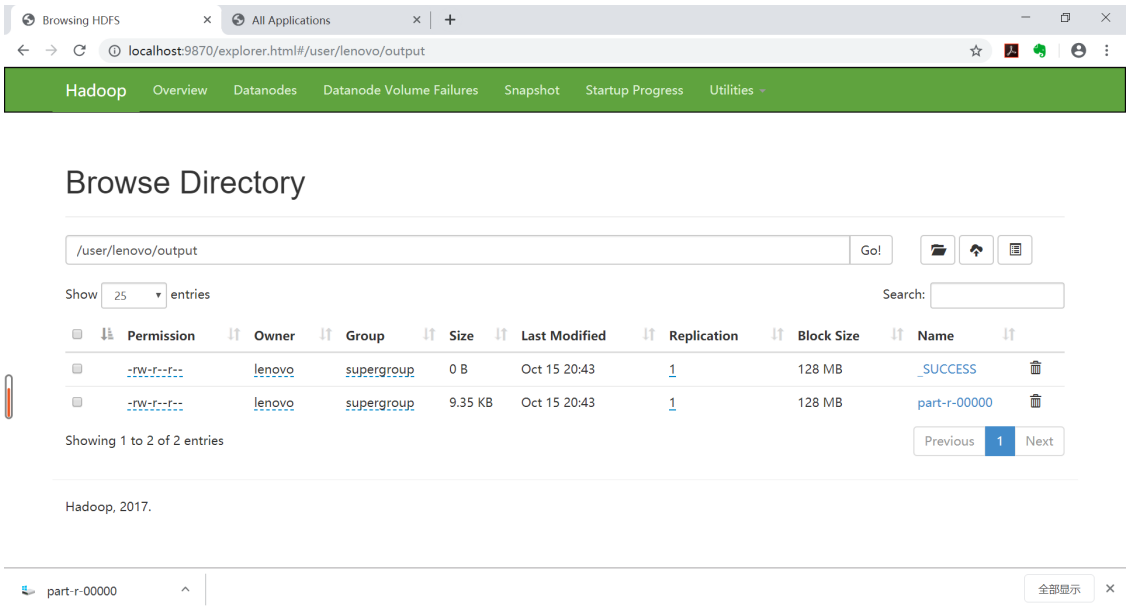
- 单机模式
  - 和src同级建立一个input文件夹，里面随便建立两个多行txt文件作为输入
  - 在Intellij菜单栏中选择Run->Edit Configurations，在弹出来的对话框中点击+，新建一个Application配置。配置Main class为WordCount（可以点击右边的...选择），Program

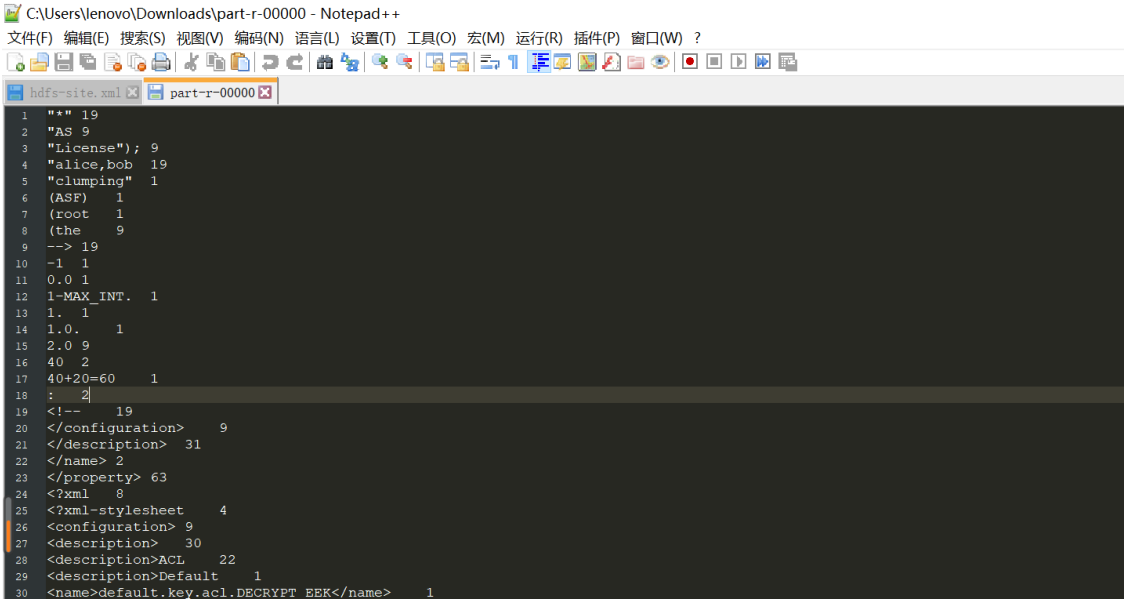
arguments为input/ output/，即输入路径为刚才创建的input文件夹，输出为output，点击ok，然后build，run就可以了

- 伪分布式
  - 命令行start-all.cmd启动hdfs
  - 在IDEA里resources中添加core-site.xml, hdfs-site.xml（hadoop文件夹里复制过来）



  - 然后运行即可，结果可以从hdfs的web页面下载





- 注意由于hadoop设定，下次运行前务必删除output文件夹

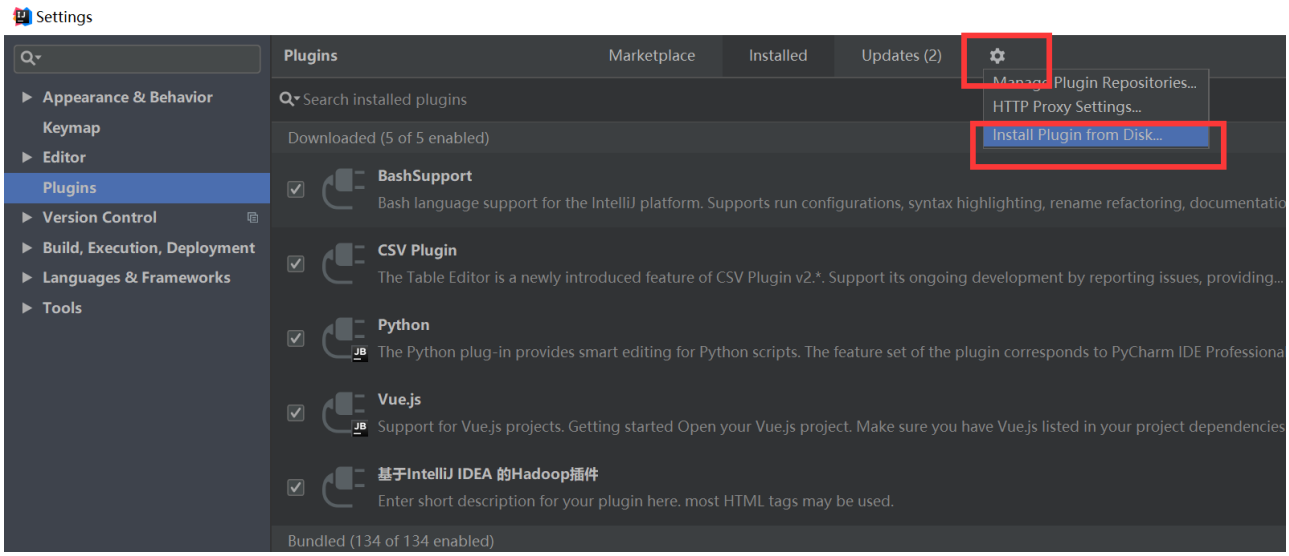
Intellij IDEA配置HDFS可视化插件

1. 下载HadoopIntellijPlugin
   - github地址：https://github.com/fangyuzhong2016/HadoopIntellijPlugin **Github**上的源码先编译再用 网上也有别人编译好的zip但是要收费很不爽
   - 编译：如果是要开发则需要新建IDEA Plugin项目，但是编译只要将github导入maven中，修改pom.xml（主要是hadoop版本改成自己的，IDEA安装目录改成自己的，就可以了。然后

```
mvn clean
mvn assembly:assembly
```

   第一次出错，重新clean一下好了，找到target下的.zip文件为编译好的插件安装包，无需解压。

2. 安装HadoopIntellijPlugin File-Settings-Editor-Plugins

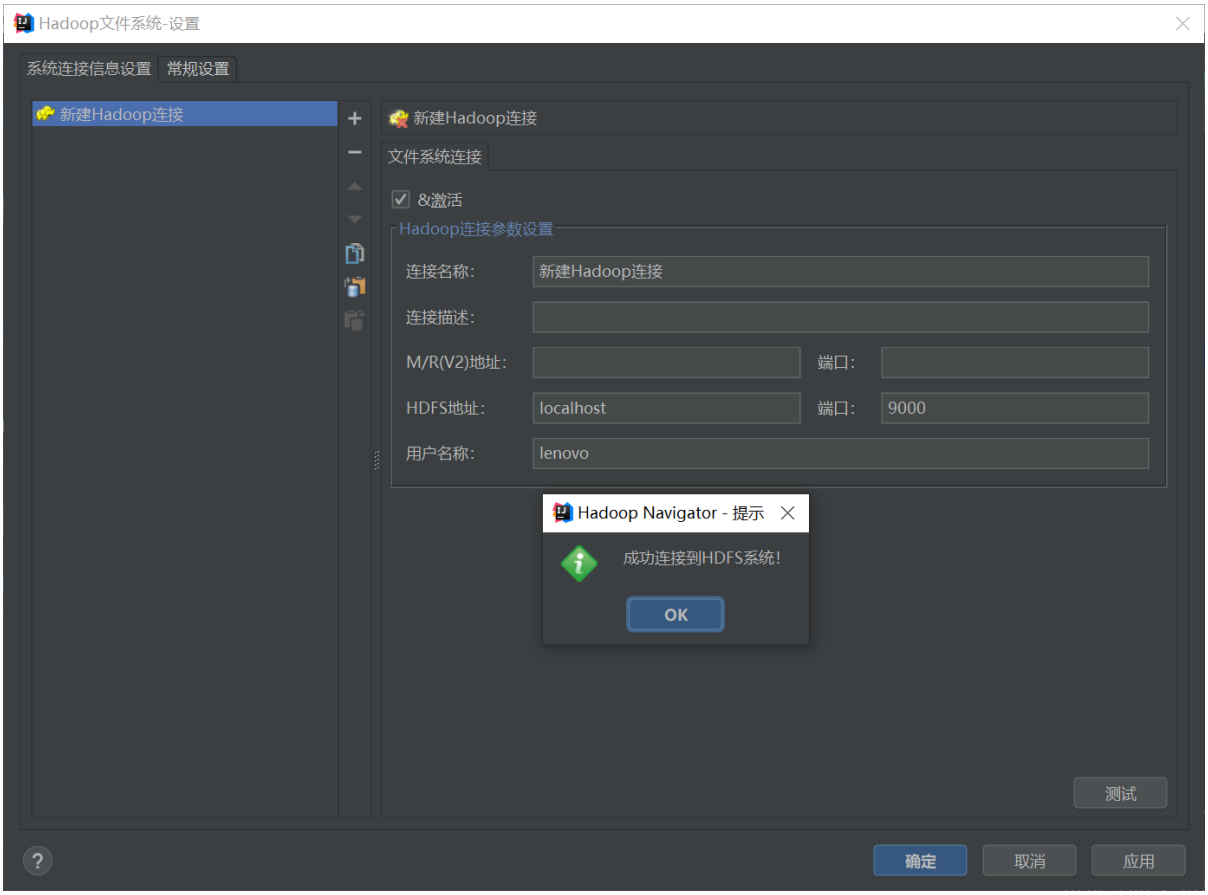


3. 按提示要重启IDEA

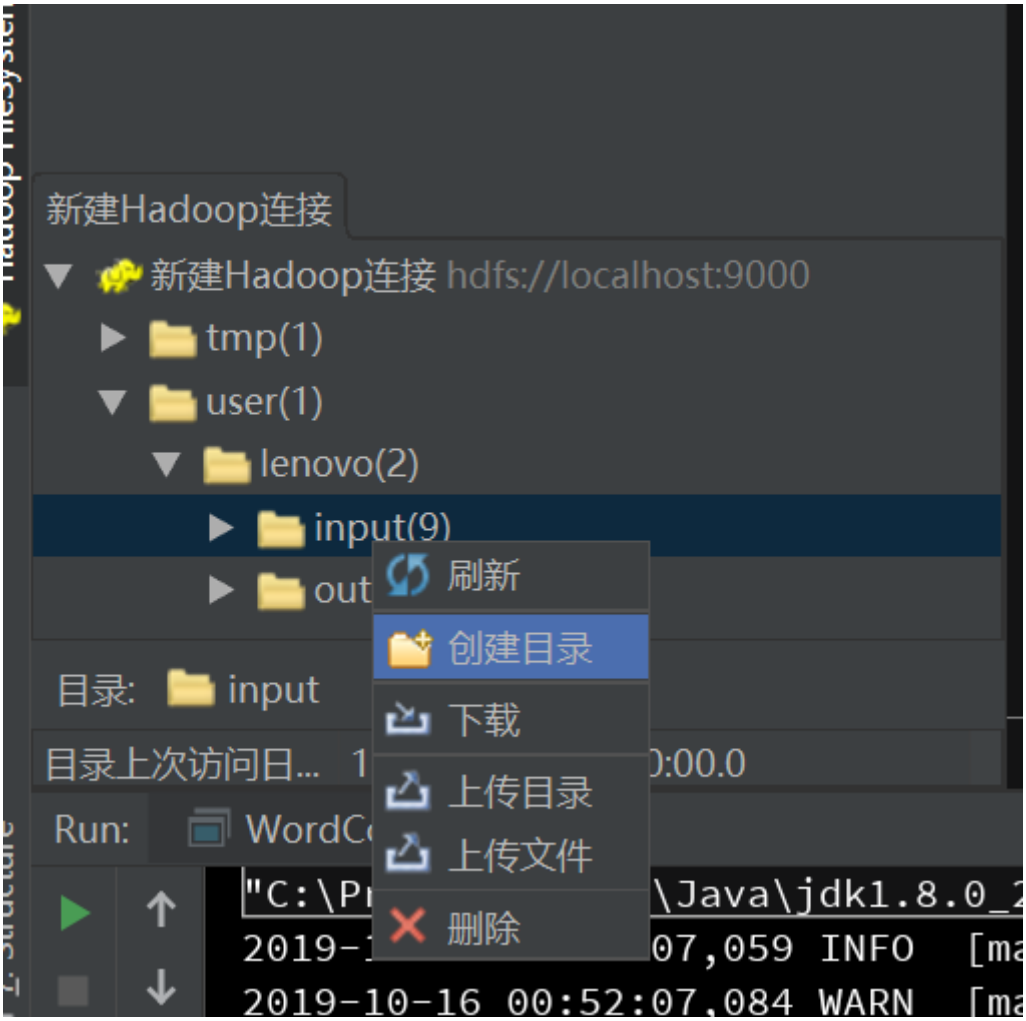
- （不要忘了关命令行java否则又启动不起来了）

4. 使用HadoopIntellijPlugin

- 两个地方进入
- 新建连接，点击左栏边的 + 号，填写正确的HDFS地址和端口



- 成功口可以看到界面



这样码代码的环境就比较宜人了哈哈哈哈哈