1 环境准备

在虚拟机安装Cent OS 6.x的操作系统（我这里是CentOS 6.6）。

1.1 修改主机名

修改主机名为spark1234，并修改配置文件确保重启后主机名依然生效。
在配置文件/etc/sysconfig/network修改主机名为spark1234。

```
# hostname spark1234
# vim /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=spark1234
```

配置主机名解析：
在/etc/hosts配置主机ip和主机名的解析

```
[root@spark1234 ~]# vim /etc/hosts

127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.9.14 spark1234
```

1.2 关闭防火墙

关闭iptables：

```
# service iptables stop
# chkconfig iptables off
# chkconfig --list | grep iptables
iptables        0:关闭    1:关闭    2:关闭    3:关闭    4:关闭    5:关闭    6:关闭

```

关闭selinux：

```
# setenforce 0
# vim /etc/sysconfig/selinux
```

```
[root@spark1234 ~]# vim /etc/sysconfig/selinux

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
#     targeted - Targeted processes are protected,
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

1.3 安装jdk

1）首先卸载openjdk

```
--查看java版本
[root@spark1234 ~]# java -version
java version "1.7.0_45"
OpenJDK Runtime Environment (rhel-2.4.3.3.el6-x86_64 u45-b15)
OpenJDK 64-Bit Server VM (build 24.45-b08, mixed mode)

--查看安装源
[root@spark1234 ~]# rpm -qa | grep java
java-1.7.0-openjdk-1.7.0.45-2.4.3.3.el6.x86_64
tzdata-java-2013g-1.el6.noarch
java-1.6.0-openjdk-1.6.0.0-1.66.1.13.0.el6.x86_64

-- 卸载
[root@spark1234 ~]# rpm -e --nodeps java-1.7.0-openjdk-1.7.0.45-2.4.3.3.el6.x86_64
[root@spark1234 ~]# rpm -e --nodeps tzdata-java-2013g-1.el6.noarch
[root@spark1234 ~]# rpm -e --nodeps java-1.6.0-openjdk-1.6.0.0-
1.66.1.13.0.el6.x86_64

```

```
--验证是否卸载成功
[root@spark1234 ~]# rpm -qa | grep java
[root@spark1234 ~]# java -version
-bash: /usr/bin/java: 没有那个文件或目录
```

2）安装java

```
-- 下载并解压java源码包
[root@spark1234 java]# mkdir /usr/local/java
[root@spark1234 java]# mv jdk-7u79-linux-x64.tar.gz /usr/local/java
[root@spark1234 java]# cd /usr/local/java
[root@spark1234 java]# tar xvf jdk-7u79-linux-x64.tar.gz
[root@spark1234 java]# ls
jdk1.7.0_79  jdk-7u79-linux-x64.tar.gz
[root@spark1234 java]#
```

3）设置环境变量

```
[root@spark1234 java]# vim /etc/profile
[root@spark1234 java]# tail /etc/profile
export JAVA_HOME=/usr/local/java/jdk1.7.0_79
export JRE_HOME=/usr/local/java/jdk1.7.0_79/jre
export
CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar:$JRE_HOME/lib:$CLASSPAT
H
export PATH=$JAVA_HOME/bin:$PATH

-- 生效环境变量
[root@spark1234 ~]# source /etc/profile
```

4）验证

```
-- 验证
[root@spark1234 ~]# java -version
java version "1.7.0_79"
Java(TM) SE Runtime Environment (build 1.7.0_79-b15)
Java HotSpot(TM) 64-Bit Server VM (build 24.79-b02, mixed mode)
[root@spark1234 ~]# javac -version
javac 1.7.0_79
```

1.4 创建hadoop用户并配置免密码认证

（1）、创建hadoop用户

```
-- 创建hadoop用户
[root@spark1234 ~]# useradd hadoop
-- 设置hadoop用户的密码为hadoop
[root@spark1234 ~]# echo "hadoop" | passwd --stdin hadoop
```

（2）配置免密码登陆

如果不配置密码也可以，但是每次启动hadoop服务都需要输入密码，建议配置

```
-- 切换到hadoop用户
[root@spark1234 ~]# su - hadoop

-- 生成公钥，  一路回车
[hadoop@spark1234 ~]$ /usr/bin/ssh-keygen -t rsa -N ""

-- 将公钥内容写入文件authorized_keys
[hadoop@spark1234 ~]$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

-- 验证authorized_keys文件的权限，  必须是644，  如果不是，  需要修改，  否则免密失败
[hadoop@spark1234 ~]$ ll ~/.ssh/authorized_keys
    -rw-rw-r-- 1 hadoop hadoop 398 11月  14 21:32 /home/hadoop/.ssh/authorized_keys
-- 修改authorized_keys文件的权限
[hadoop@spark1234 ~]$ chmod 644 ~/.ssh/authorized_keys

-- 验证免密
[hadoop@spark1234 ~]$ ssh spark1234
Last login: Tue Nov 14 21:33:59 2017 from spark1234
[hadoop@spark1234 ~]$
```

2. 大数据相关组件安装

2.1 软件清单

软件	版本
hadoop	hadoop-2.6.0-cdh5.6.0
hive	hive-1.1.0-cdh5.6.0
spark	spark-1.6.3-bin-2.6.0-cdh5.6.0
mysql	mysql-5.6

（1）hadoop、hive、spark可从cdh的官网下载，链接：http://archive.cloudera.com/cdh5/cdh/5/

（2）mysql采用rpm包安装

下载链接：https://cdn.mysql.com//Downloads/MySQL-5.6/MySQL-5.6.38-1.el6.x86_64.rpm-bundle.tar

2.2 hadoop安装

（1）解压安装包

```
$ tar -xzf hadoop-2.6.0-cdh5.6.0.tar.gz -C /home/hadoop/app/
```

（2）设置环境变量

在hadoop用户下，在~/.bashrc文件增加两行：

```
[hadoop@spark1234 ~]$ vim  ~/.bashrc
增加如下两行内容：
export HADOOP_HOME=/home/hadoop/app/hadoop-2.6.0-cdh5.6.0
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
```

```
[hadoop@spark1234 ~]$ cat ~/.bashrc
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# User specific aliases and functions
export HADOOP_HOME=/home/hadoop/app/hadoop-2.6.0-cdh5.6.0
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
[hadoop@spark1234 ~]$
```

生效：

```
[hadoop@spark1234 ~]$ source ~/.bashrc
```

（3）、修改core-site.xml文件
切换到目录： $HADOOP_HOME/etc/hadoop
编辑core-site.xml， 增加配置：

```
<configuration>
<property>
    <name>fs.defaultFS</name>
    <value>hdfs://spark1234:8020</value>
</property>
<property>
    <name>hadoop.tmp.dir</name>
    <value>/home/hadoop/app/hadoop-2.6.0-cdh5.6.0/tmp</value>
</property>
</configuration>
```

新建配置文件里面配置的目录：
一定要在hadoop用户下创建， 或者hadoop用户有权限操作这个目录

```
[hadoop@spark1234 ~]# mkdir /home/hadoop/app/hadoop-2.6.0-cdh5.6.0/tmp
```

（4）修改配置文件hdfs-site.xml
切换到目录： $HADOOP_HOME/etc/hadoop
编辑hdfs-site.xml， 增加配置：

```
<configuration>
<property>
    <name>dfs.replication</name>
    <value>1</value>
</property>
</configuration>
```

（5）修改配置文件hadoop-env.sh

切换到目录： $HADOOP_HOME/etc/hadoop

修改JAVA_HOME的配置：

export JAVA_HOME=${JAVA_HOME} 修改为： export JAVA_HOME=/usr/local/java/jdk1.7.0_79

```
# optional.  When running a distributed configuration it
# set JAVA_HOME in this file, so that it is correctly def
# remote nodes.

# The java implementation to use.
export JAVA_HOME=/usr/local/java/jdk1.7.0_79

# The jsvc implementation to use. Jsvc is required to run
# that bind to privileged ports to provide authentication
```

（6）修改配置文件mapred-site.xml

切换到目录： $HADOOP_HOME/etc/hadoop

```
[hadoop@spark1234 hadoop]$ cp mapred-site.xml.template mapred-site.xml
[hadoop@spark1234 hadoop]$
[hadoop@spark1234 hadoop]$ vim mapred-site.xml
```

编辑mapred-site.xml， 增加配置：

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

（7）修改配置文件yarn-site.xml

增加如下配置：

```
<configuration>
    <property>
        <name>yarn.log-aggregation-enable</name>
        <value>true</value>
    </property>
```

```
    <property>
         <name>yarn.log.server.url</name>
         <value>http://spark1234:19888/jobhistory/job/</value>
    </property>
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>
</configuration>
```

（8）格式化

```
[hadoop@spark1234 ~]$ hdfs namenode -format
```

```
.7/11/14 23:33:40 INFO namenode.FSNamesystem: dfs.namenode.safemode.threshold-pct = 0.9990000128746033
.7/11/14 23:33:40 INFO namenode.FSNamesystem: dfs.namenode.safemode.min.datanodes = 0
.7/11/14 23:33:40 INFO namenode.FSNamesystem: dfs.namenode.safemode.extension     = 30000
.7/11/14 23:33:40 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 10
.7/11/14 23:33:40 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
.7/11/14 23:33:40 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
.7/11/14 23:33:40 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
.7/11/14 23:33:40 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time is 600000 millis
.7/11/14 23:33:40 INFO util.GSet: Computing capacity for map NameNodeRetryCache
.7/11/14 23:33:40 INFO util.GSet: VM type       = 64-bit
.7/11/14 23:33:40 INFO util.GSet: 0.029999999329447746% max memory 966.7 MB = 297.0 KB
.7/11/14 23:33:40 INFO util.GSet: capacity      = 2^15 = 32768 entries
.7/11/14 23:33:40 INFO namenode.NNConf: ACLs enabled? false
.7/11/14 23:33:40 INFO namenode.NNConf: XAttrs enabled? true
.7/11/14 23:33:40 INFO namenode.NNConf: Maximum size of an xattr: 16384
.7/11/14 23:33:40 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1078578980-192.168.9.14-1510673620265
.7/11/14 23:33:40 INFO common.Storage: Storage directory /home/hadoop/app/hadoop-2.6.0-cdh5.6.0/tmp/dfs/name has been successfully formatted.
.7/11/14 23:33:40 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
.7/11/14 23:33:40 INFO util.ExitUtil: Exiting with status 0
.7/11/14 23:33:40 INFO namenode.NameNode: SHUTDOWN_MSG:
'************************************************************
HUTDOWN_MSG: Shutting down NameNode at spark1234/192.168.9.14
************************************************************/
```

（9）启动服务

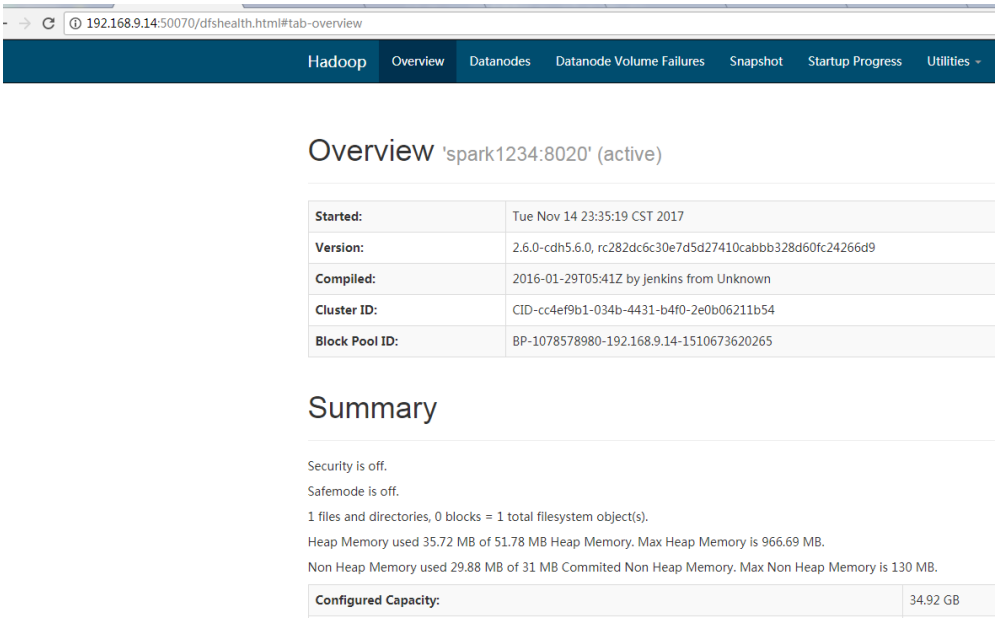
```
-- 启动hdfs服务
[hadoop@spark1234 ~]$ start-dfs.sh
-- 启动yarn服务
[hadoop@spark1234 ~]$ start-yarn.sh
-- 验证服务
[hadoop@spark1234 ~]$ jps
```

使用jps查看进程：

```
[hadoop@spark1234 ~]$ jps
8040 NameNode
8328 SecondaryNameNode
8133 DataNode
8470 ResourceManager
8569 NodeManager
8882 Jps
```

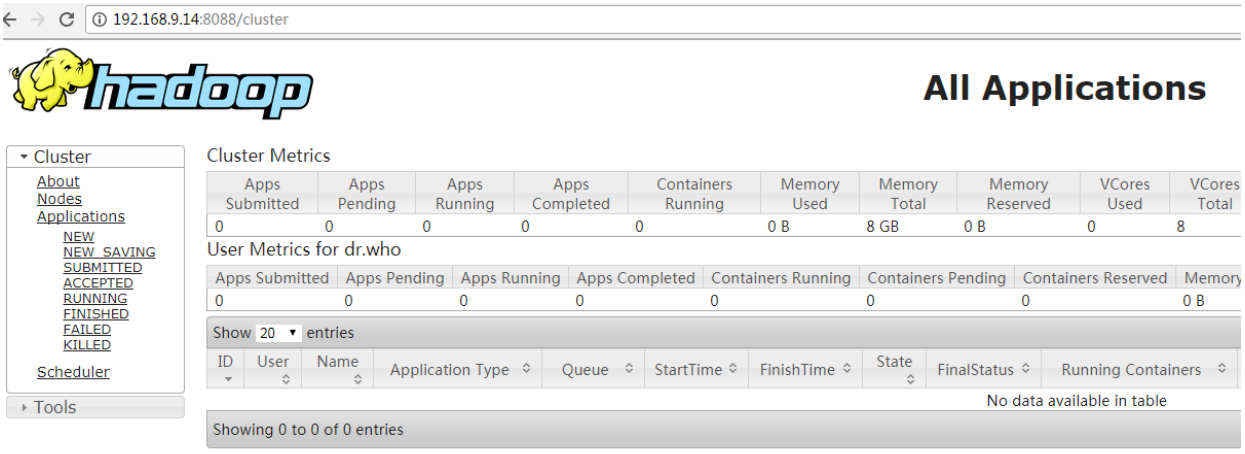
查看hdfs的web ui：

http://192.168.9.14:50070/dfshealth.html#tab-overview



查看yarn的web ui：

http://192.168.9.14:8088/cluster



2.3 MySQL安装

（1）卸载现有的mysql

```
[root@spark1234 ~]# rpm -qa | grep mysql
mysql-5.1.73-3.el6_5.x86_64
mysql-libs-5.1.73-3.el6_5.x86_64
mysql-devel-5.1.73-3.el6_5.x86_64
[root@spark1234 ~]#
[root@spark1234 ~]# rpm -e --nodeps mysql-5.1.73-3.el6_5.x86_64
[root@spark1234 ~]# rpm -e --nodeps mysql-libs-5.1.73-3.el6_5.x86_64
[root@spark1234 ~]# rpm -e --nodeps mysql-devel-5.1.73-3.el6_5.x86_64

```

（2）、将下载的MySQL-5.6.38-1.el6.x86_64.rpm-bundle.tar上次到主机

```
--解压
# tar -xf MySQL-5.6.38-1.el6.x86_64.rpm-bundle.tar

[root@spark1234 test]# ll
总用量 445708
-rw-r--r-- 1 root root  228198400 11月 14 13:52 MySQL-5.6.38-1.el6.x86_64.rpm-bundle.tar
-rw-r--r-- 1 7155 31415  19045216 9月  14 19:00 MySQL-client-5.6.38-1.el6.x86_64.rpm
-rw-r--r-- 1 7155 31415   3391312 9月  14 19:00 MySQL-devel-5.6.38-1.el6.x86_64.rpm
-rw-r--r-- 1 7155 31415  90407828 9月  14 19:01 MySQL-embedded-5.6.38-1.el6.x86_64.rpm
-rw-r--r-- 1 7155 31415  57551568 9月  14 19:01 MySQL-server-5.6.38-1.el6.x86_64.rpm
-rw-r--r-- 1 7155 31415   1964616 9月  14 19:01 MySQL-shared-5.6.38-1.el6.x86_64.rpm
-rw-r--r-- 1 7155 31415   3969744 9月  14 19:01 MySQL-shared-compat-5.6.38-1.el6.x86_64.rpm
-rw-r--r-- 1 7155 31415  51861916 9月  14 19:01 MySQL-test-5.6.38-1.el6.x86_64.rpm

-- 安装
[root@spark1234 test]# rpm -ivh MySQL*.rpm

-- 修改配置文件的位置
[root@spark1234 test]# cp /usr/share/mysql/my-default.cnf /etc/my.cnf

```

（3）初始化MySQL并设置密码

```

```

```
-- 初始化mysql
[root@spark1234 test]# /usr/bin/mysql_install_db

-- 启动mysql
[root@spark1234 test]# service mysql start

-- 查看root账号的密码
[root@spark1234 test]# cat /root/.mysql_secret
# The random password set for the root user at Tue Nov 14 22:01:34 2017 (local time): KAxcgzc3R_qVJTB7

-- 进入MySQL，  并修改密码
[root@spark1234 test]# mysql -uroot -pKAxcgzc3R_qVJTB7

-- 设置初始密码为123456
mysql> SET PASSWORD = PASSWORD('123456');
Query OK, 0 rows affected (0.03 sec)

mysql> exit

```

设置权限：

```
mysql> grant all privileges on *.* to root@'localhost' identified by '123456' with grant option;
mysql> grant all privileges on *.* to root@'127.0.0.1' identified by '123456' with grant option;
mysql> grant all privileges on *.* to root@'%' identified by '123456' with grant option;
mysql> flush privileges;
```

（4）设置开机启动

```
[root@spark1234 test]# chkconfig mysql on
[root@spark1234 test]# chkconfig --list | grep mysql
mysql           0:关闭    1:关闭    2:启用    3:启用    4:启用    5:启用    6:关闭
```

2.4 hive的安装

（1）解压安装包

```
$ tar -xzf hive-1.1.0-cdh5.6.0.tar.gz -C /home/hadoop/app/
```

（2）设置环境变量

在hadoop用户下，在~/.bashrc文件增加两行：

```
[hadoop@spark1234 ~]$ vim  ~/.bashrc
增加如下两行内容：
export HIVE_HOME=/home/hadoop/app/hive-1.1.0-cdh5.6.0
export PATH=$HIVE_HOME/bin:$PATH
```

```
[hadoop@spark1234 ~]$ cat .bashrc
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# User specific aliases and functions
export HADOOP_HOME=/home/hadoop/app/hadoop-2.6.0-cdh5.6.0
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
export HIVE_HOME=/home/hadoop/app/hive-1.1.0-cdh5.6.0
export PATH=$HIVE_HOME/bin:$PATH
[hadoop@spark1234 ~]$
[hadoop@spark1234 ~]$
```

生效：

```
[hadoop@spark1234 ~]$ source .bashrc
```

（3）、在mysql创建hive使用的表

```
-- 创建hive数据库
mysql> create database hive;
-- 设置编码，  一定要设置成latin1，  否则hive建表和删表会卡住
mysql> alter database hive character set latin1;

```

```
[hadoop@spark1234 ~]$
[hadoop@spark1234 ~]$ mysql -uroot -p123456
Warning: Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 5.6.38 MySQL Community Server (GPL)

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database hive;
Query OK, 1 row affected (0.06 sec)

mysql> alter database hive character set latin1;
Query OK, 1 row affected (0.00 sec)

mysql>
```

（4）修改hive的配置文件

在 $HIVE_HOME/conf 目录创建文件hive-site.xml

配置如下：

```xml
<?xml version="1.0" encoding="UTF-8"?>

<!--Autogenerated by Cloudera Manager-->
<configuration>
    <property>
        <name>javax.jdo.option.ConnectionURL</name>
        <value>jdbc:mysql://127.0.0.1:3306/hive?createDatabaseIfNotExist=true</value>
  </property>

    <property>
        <name>javax.jdo.option.ConnectionDriverName</name>
        <value>com.mysql.jdbc.Driver</value>
    </property>

    <property>
        <name>javax.jdo.option.ConnectionUserName</name>
        <value>root</value>
    </property>

    <property>
       <name>javax.jdo.option.ConnectionPassword</name>
       <value>123456</value>
    </property>
</configuration>
```

（5）将mysql的jdk驱动包复制到hive的lib目录下

```
$ cp mysql-connector-java-5.1.44-bin.jar $HIVE_HOME/lib/
```

（6）进入hive

进入hive，并验证可用：

```
[hadoop@spark1234 soft]$ hive
hive> show tables;
hive> create table test(id int, name string);
hive> insert into test values (1, "abc");
hive> select * from test;
```

```
[hadoop@spark1234 soft]$ hive
which: no hbase in (/home/hadoop/app/hive-1.1.0-cdh5.6.0/bin:/home/hadoop/app/hadoop-2.6.0-
in:/home/hadoop/app/hadoop-2.6.0-cdh5.6.0/bin:/home/hadoop/app/hadoop-2.6.0-cdh5.6.0/sbin:/
usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/home/hadoop/bin)

Logging initialized using configuration in jar:file:/home/hadoop/app/hive-1.1.0-cdh5.6.0/li
s

WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive>
    > show tables;
OK
Time taken: 1.127 seconds
hive> create table test(id int, name string);
OK
Time taken: 1.173 seconds
```

```
hive>
    > insert into test values (1, "abc");
Query ID = hadoop_20171114235050_aacb55f1-1786-46bf-8880-dbdde924630e
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1510673764000_0001, Tracking URL = http://spark1234:8088/pro
Kill Command = /home/hadoop/app/hadoop-2.6.0-cdh5.6.0/bin/hadoop job  -kill job
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2017-11-14 23:50:33,599 Stage-1 map = 0%,  reduce = 0%
2017-11-14 23:50:48,872 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.91 s
MapReduce Total cumulative CPU time: 1 seconds 910 msec
Ended Job = job_1510673764000_0001
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to: hdfs://spark1234:8020/user/hive/warehouse/test/.hive-staging_hi
Loading data to table default.test
Table default.test stats: [numFiles=1, numRows=1, totalSize=6, rawDataSize=5]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 1.91 sec   HDFS Read: 3421 HDFS Write:
Total MapReduce CPU Time Spent: 1 seconds 910 msec
OK
Time taken: 45.14 seconds
hive>
    > select * from test;
OK
1       abc
Time taken: 0.43 seconds, Fetched: 1 row(s)
hive>
```

2.5 spark安装

（1）解压安装包

```
$ tar -xzf spark-1.6.3-bin-2.6.0-cdh5.6.0.tgz -C /home/hadoop/app/
```

（2）设置环境变量

在hadoop用户下，在~/.bashrc文件增加两行：

```
export SPARK_HOME=/home/hadoop/app/spark-1.6.3-bin-2.6.0-cdh5.6.0
export PATH=$SPARK_HOME/bin:$SPARK_HOME/sbin:$PATH
```

```
[hadoop@spark1234 ~]$ cat .bashrc
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# User specific aliases and functions
export HADOOP_HOME=/home/hadoop/app/hadoop-2.6.0-cdh5.6.0
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
export HIVE_HOME=/home/hadoop/app/hive-1.1.0-cdh5.6.0
export PATH=$HIVE_HOME/bin:$PATH
export SPARK_HOME=/home/hadoop/app/spark-1.6.3-bin-2.6.0-cdh5.6.0
export PATH=$SPARK_HOME/bin:$SPARK_HOME/sbin:$PATH
[hadoop@spark1234 ~]$
```

环境变量生效：

```
[hadoop@spark1234 ~]$ source .bashrc
```

（3）修改配置文件

进入目录： $SPARK_HOME/conf目录

```
[hadoop@spark1234 conf]$ cp spark-env.sh.template spark-env.sh
```

增加指定hadoop配置文件的配置：

export HADOOP_CONF_DIR=/home/hadoop/app/hadoop-2.6.0-cdh5.6.0/etc/hadoop

```
# Generic options for the daemons used in the standalone deploy mode
# - SPARK_CONF_DIR      Alternate conf dir. (Default: ${SPARK_HOME}/conf)
# - SPARK_LOG_DIR       Where log files are stored.  (Default: ${SPARK_HOME}/
# - SPARK_PID_DIR       Where the pid file is stored. (Default: /tmp)
# - SPARK_IDENT_STRING  A string representing this instance of spark. (Defaul
# - SPARK_NICENESS      The scheduling priority for daemons. (Default: 0)
export HADOOP_CONF_DIR=/home/hadoop/app/hadoop-2.6.0-cdh5.6.0/etc/hadoop
[hadoop@spark1234 conf]$
```

（4）将hive的配置文件复制到spark的配置目录下
这样spark就可以直接读取hive中的表

```
[hadoop@spark1234 conf]$ cp $HIVE_HOME/conf/hive-site.xml  $SPARK_HOME/conf/
```

（5）进入spark

```
$ spark-shell --master yarn --jars /home/hadoop/app/hive-1.1.0-cdh5.6.0/lib/mysql-connector-java-5.1.44-bin.jar
```

```
have its own datastore table.
17/11/15 00:33:59 INFO DataNucleus.Query: Reading in results for query "org.datanucleus.store.rdbms.query.SQLQ
s closing
17/11/15 00:33:59 INFO metastore.MetaStoreDirectSql: Using direct SQL, underlying DB is MYSQL
17/11/15 00:33:59 INFO metastore.ObjectStore: Initialized ObjectStore
17/11/15 00:33:59 INFO metastore.HiveMetaStore: Added admin role in metastore
17/11/15 00:33:59 INFO metastore.HiveMetaStore: Added public role in metastore
17/11/15 00:33:59 INFO metastore.HiveMetaStore: No user is added in admin role, since config is empty
17/11/15 00:33:59 INFO metastore.HiveMetaStore: 0: get_all_databases
17/11/15 00:33:59 INFO HiveMetaStore.audit: ugi=hadoop  ip=unknown-ip-addr      cmd=get_all_databases
17/11/15 00:34:00 INFO metastore.HiveMetaStore: 0: get_functions: db=default pat=*
17/11/15 00:34:00 INFO HiveMetaStore.audit: ugi=hadoop  ip=unknown-ip-addr      cmd=get_functions: db=default
17/11/15 00:34:00 INFO DataNucleus.Datastore: The class "org.apache.hadoop.hive.metastore.model.MResourceUri"
s not have its own datastore table.
17/11/15 00:34:00 INFO session.SessionState: Created local directory: /tmp/5b1487fc-e731-4b4d-8346-0c3aad8e21c
17/11/15 00:34:00 INFO session.SessionState: Created HDFS directory: /tmp/hive/hadoop/5b1487fc-e731-4b4d-8346-
17/11/15 00:34:00 INFO session.SessionState: Created local directory: /tmp/hadoop/5b1487fc-e731-4b4d-8346-0c3a
17/11/15 00:34:00 INFO session.SessionState: Created HDFS directory: /tmp/hive/hadoop/5b1487fc-e731-4b4d-8346-
17/11/15 00:34:00 INFO repl.SparkILoop: Created sql context (with Hive support)..
SQL context available as sqlContext.

scala>

scala>

scala>
```

测试读取hive中的表：

```
scala> sqlContext.sql("select * from test").show
```

```
17/11/15 00:34:54 INFO sc
+---+----+
| id|name|
+---+----+
|  1| abc|
+---+----+

scala>
```

3. 开发工具搭建

3.1 下载IDE开发工具idea

从官网下载， 如果有教育网邮箱， 可以申请免费注册使用。

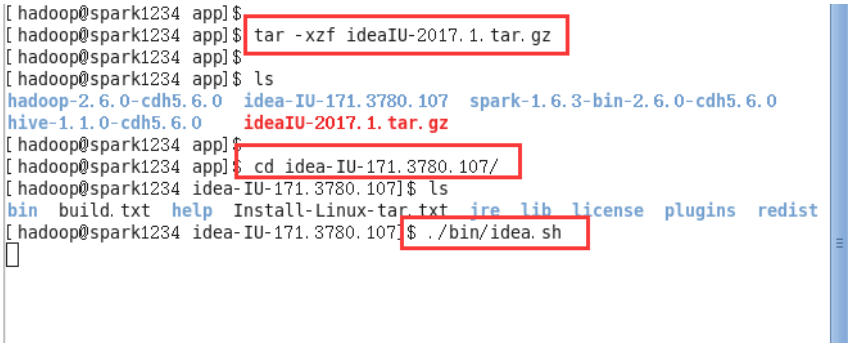
软件：

软件	版本
IDEA	idealU-2017.1
Scala的IDEA插件	scala-intellij-bin-2017.1.20

将两个软件上传到主机

3.2 安装IDEA， 配置Scala插件

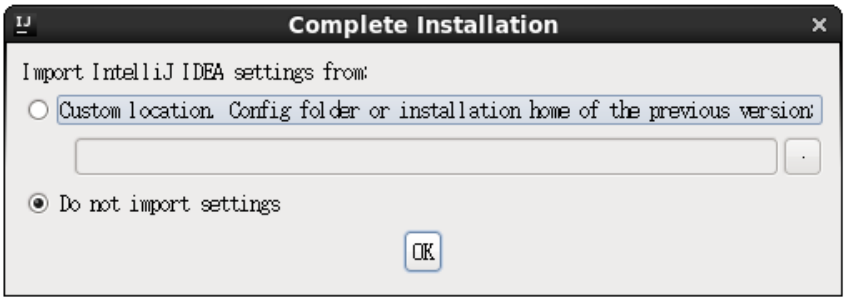
（1）启动idea

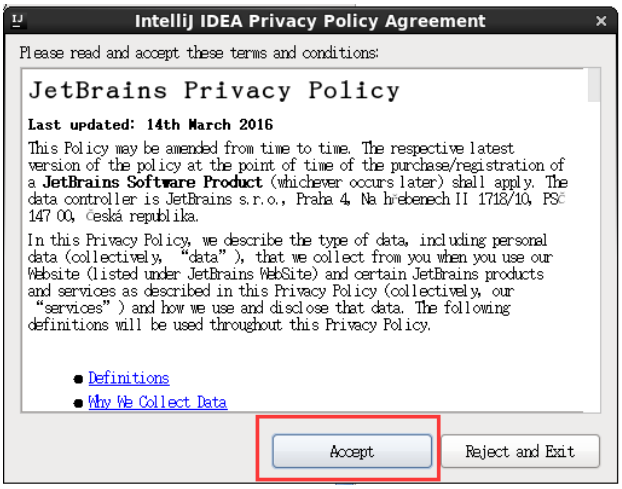


（2）配置idea向导

6.0

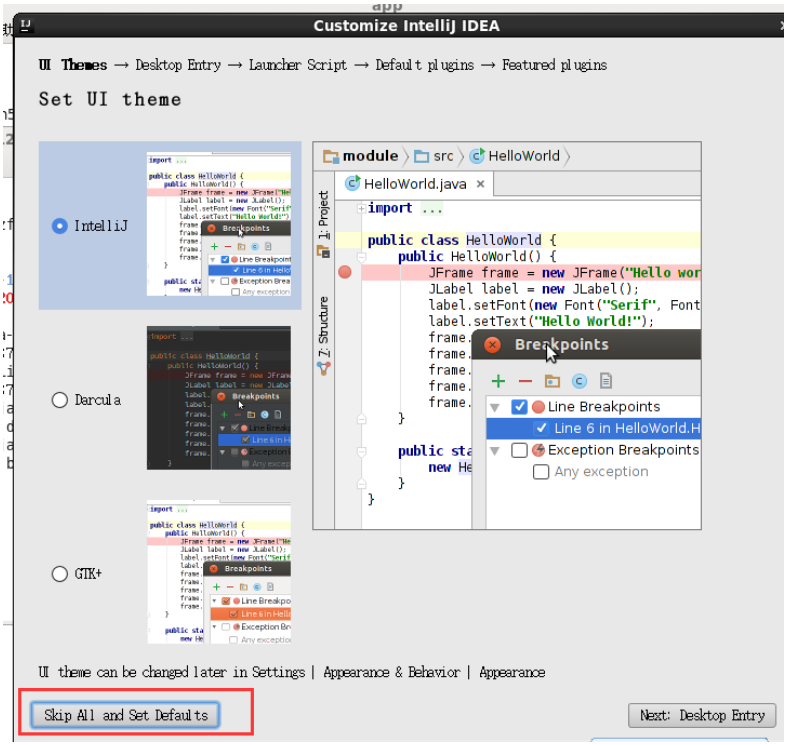
redist





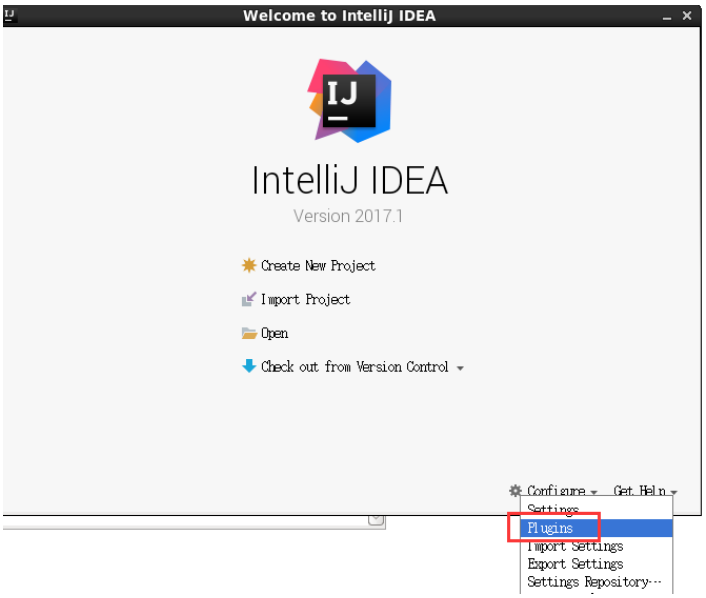
spark-1.6.3-bin-2.6.0-cdh5.6.0

ome/hadoop/.java/.userPrefs/prefs.xm

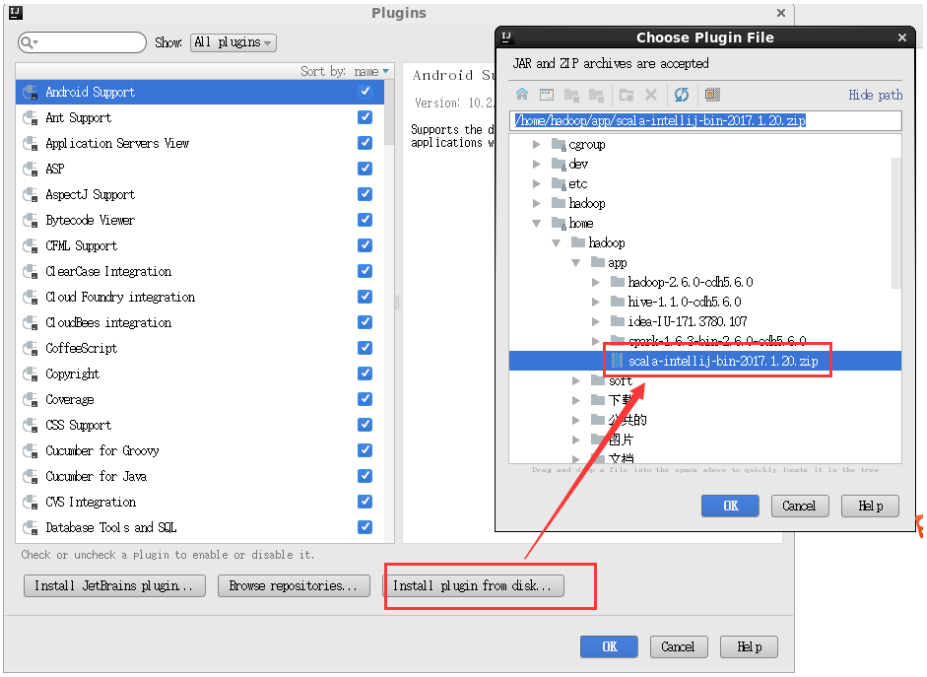


（3）配置scala插件

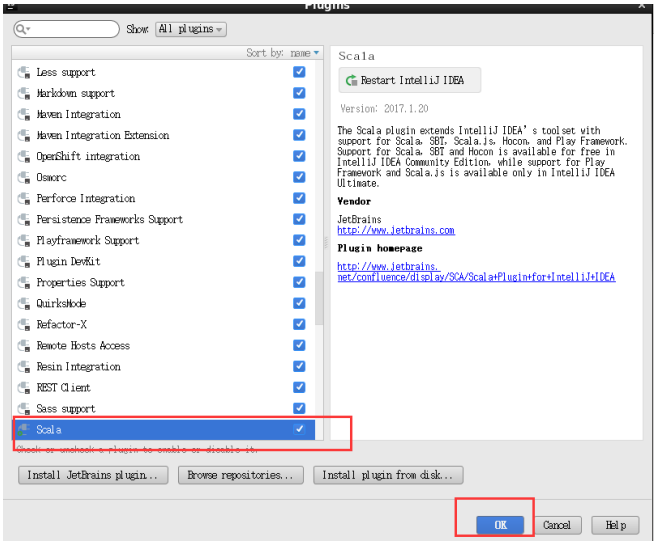
选择Plugins：



选择前面上传的scala插件压缩包：



点击OK：

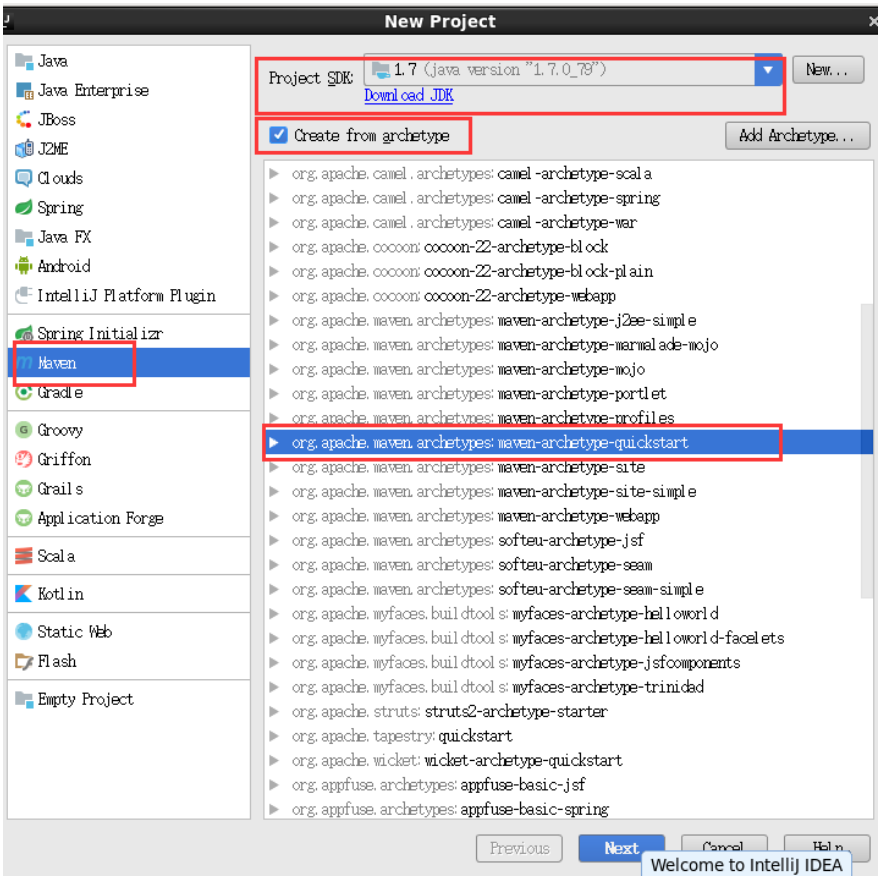


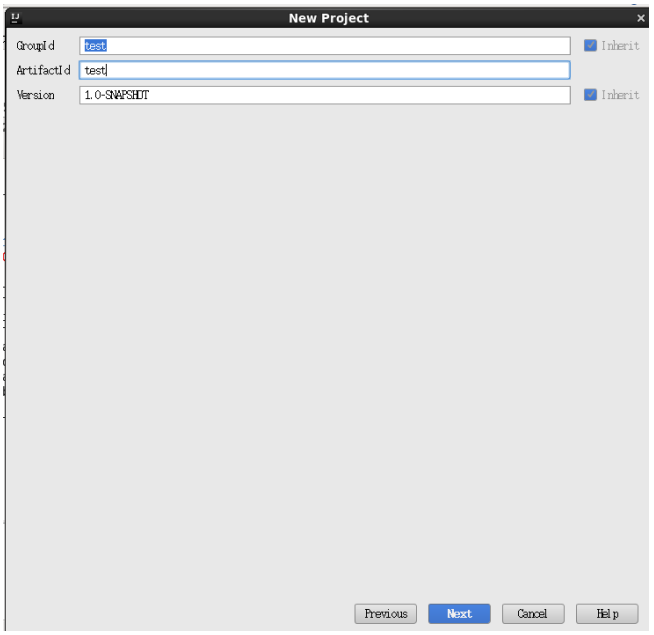
重启：

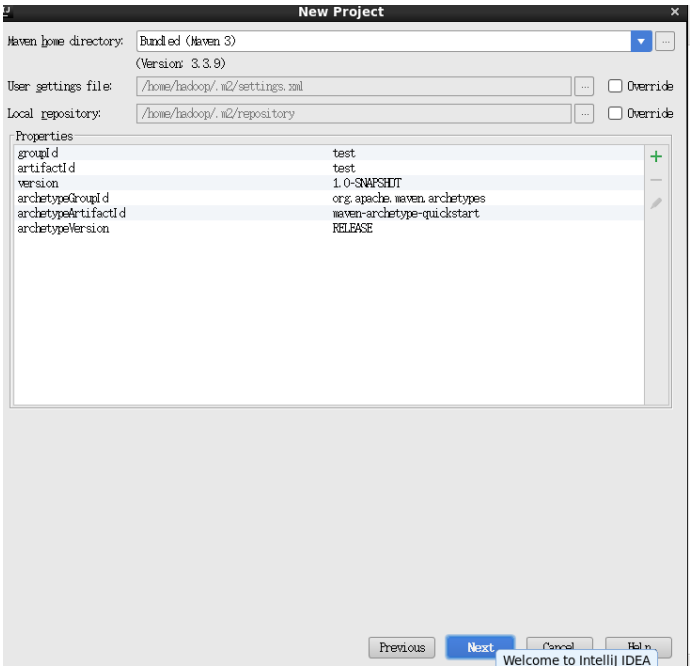
3.3 创建、配置测试的Spark项目

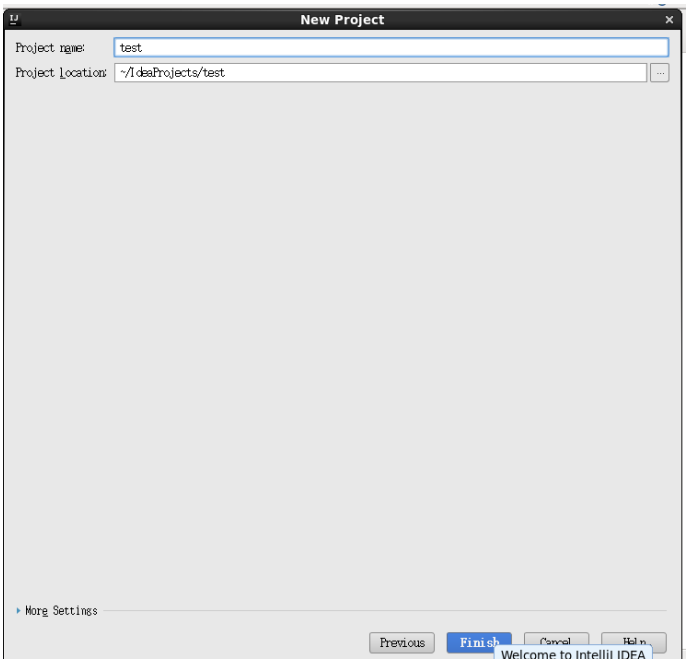
点击第一个， 创建测试项目

创建Maven工程， 第一次创建功能需要选择JDK的安装路径：


New Project
Java
Java Enterprise
JBoss
J2ME
Clouds
Spring
Java FX
Android
IntelliJ Platform Plugin
Spring Initializr
Maven
Gradle
Groovy
Griffon
Grails
Application Forge
Scala
Kotlin
Static Web
Flash
Empty Project
Project SDK:
1.7 (java version "1.7.0_79")
Download JDK
New...
Create from archetype
Add Archetype...
org.apache.camel.archetypes:camel-archetype-scala
org.apache.camel.archetypes:camel-archetype-spring
org.apache.camel.archetypes:camel-archetype-war
org.apache.cocoon:cocoon-22-archetype-block
org.apache.cocoon:cocoon-22-archetype-block-plain
org.apache.cocoon:cocoon-22-archetype-webapp
org.apache.maven.archetypes:maven-archetype-j2ee-simple
org.apache.maven.archetypes:maven-archetype-marmalade-mojo
org.apache.maven.archetypes:maven-archetype-mojo
org.apache.maven.archetypes:maven-archetype-portlet
org.apache.maven.archetypes:maven-archetype-profiles
org.apache.maven.archetypes:maven-archetype-quickstart
org.apache.maven.archetypes:maven-archetype-site
org.apache.maven.archetypes:maven-archetype-site-simple
org.apache.maven.archetypes:maven-archetype-webapp
org.apache.maven.archetypes:softeu-archetype-jsf
org.apache.maven.archetypes:softeu-archetype-seam
org.apache.maven.archetypes:softeu-archetype-seam-simple
org.apache.myfaces.buildtools:myfaces-archetype-helloworld
org.apache.myfaces.buildtools:myfaces-archetype-helloworld-facelets
org.apache.myfaces.buildtools:myfaces-archetype-jsfcomponents
org.apache.myfaces.buildtools:myfaces-archetype-trinidad
org.apache.struts:struts2-archetype-starter
org.apache.tapestry:quickstart
org.apache.wicket:wicket-archetype-quickstart
org.appfuse.archetypes:appfuse-basic-jsf
org.appfuse.archetypes:appfuse-basic-spring
Previous
Next
Cancel
Help
Welcome to IntelliJ IDEA


New Project
GroupId
test
Inherit
ArtifactId
test
Version
1.0-SNAPSHOT
Inherit
Previous
Next
Cancel
Help





（5）修改pom文件，增加项目需要的依赖

```
<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>Test</groupId>
    <artifactId>Test</artifactId>
    <version>1.0-SNAPSHOT</version>
```

```
    <packaging>jar</packaging>

    <name>JSDPI</name>
    <url>http://maven.apache.org</url>
    <repositories>
        <repository>
            <id>cloudera</id>
            <url>https://repository.cloudera.com/artifactory/cloudera-
repos/</url>
        </repository>
    </repositories>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <spark.version>1.6.0-cdh5.7.0</spark.version>
        <jedis.version>2.8.2</jedis.version>
        <hadoop.version>2.6.0-cdh5.7.0</hadoop.version>
        <fastjson.version>1.2.14</fastjson.version>
        <jetty.version>9.2.5.v20141112</jetty.version>
        <container.version>2.17</container.version>
        <java.version>1.8</java.version>
        <scala.version>2.10.6</scala.version>
    </properties>



    <dependencies>
        <dependency>
            <groupId>redis.clients</groupId>
            <artifactId>jedis</artifactId>
            <version>${jedis.version}</version>
        </dependency>
        <dependency>
            <groupId>org.apache.hadoop</groupId>
            <artifactId>hadoop-common</artifactId>
            <version>${hadoop.version}</version>
            <exclusions>
                <exclusion>
                    <groupId>javax.servlet</groupId>
                    <artifactId>*</artifactId>
                </exclusion>
            </exclusions>
        </dependency>
        <dependency>
            <groupId>org.apache.hadoop</groupId>
            <artifactId>hadoop-hdfs</artifactId>
```

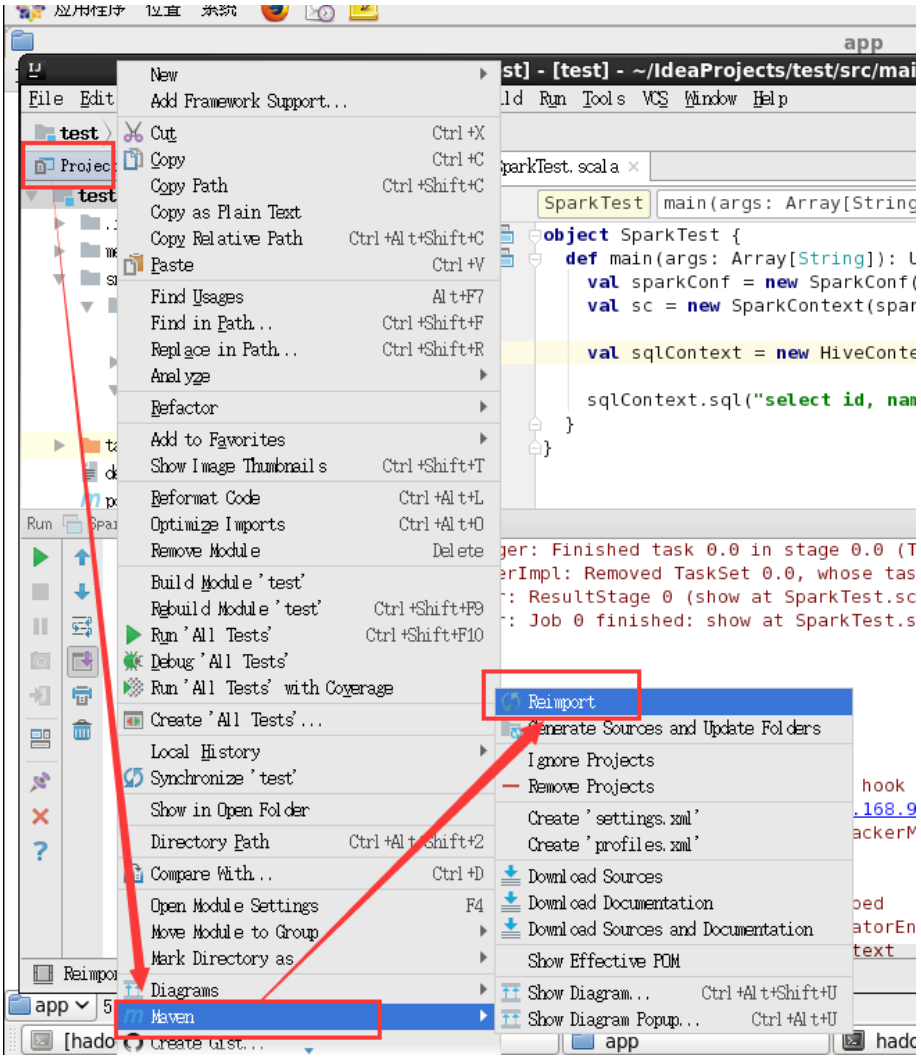
```
            <version>${hadoop.version}</version>
            <exclusions>
                <exclusion>
                    <groupId>javax.servlet</groupId>
                    <artifactId>*</artifactId>
                </exclusion>
            </exclusions>
        </dependency>
        <dependency>
            <groupId>org.apache.hadoop</groupId>
            <artifactId>hadoop-client</artifactId>
            <version>${hadoop.version}</version>
            <exclusions>
                <exclusion>
                    <groupId>javax.servlet</groupId>
                    <artifactId>*</artifactId>
                </exclusion>
            </exclusions>
        </dependency>
        <dependency>
            <groupId>org.apache.spark</groupId>
            <artifactId>spark-core_2.10</artifactId>
            <version>${spark.version}</version>
        </dependency>
        <dependency>
            <groupId>org.apache.spark</groupId>
            <artifactId>spark-sql_2.10</artifactId>
            <version>${spark.version}</version>
        </dependency>
        <dependency>
            <groupId>org.apache.spark</groupId>
            <artifactId>spark-streaming_2.10</artifactId>
            <version>${spark.version}</version>
        </dependency>
        <dependency>
            <groupId>org.apache.spark</groupId>
            <artifactId>spark-yarn_2.10</artifactId>
            <version>${spark.version}</version>
        </dependency>
        <dependency>
            <groupId>org.apache.spark</groupId>
            <artifactId>spark-hive_2.10</artifactId>
            <version>${spark.version}</version>
        </dependency>
        <dependency>
            <groupId>com.google.guava</groupId>
```

```
            <artifactId>guava</artifactId>
            <version>18.0</version>
        </dependency>
        <dependency>
            <groupId>com.alibaba</groupId>
            <artifactId>fastjson</artifactId>
            <version>${fastjson.version}</version>
        </dependency>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>org.spark-project.hive</groupId>
            <artifactId>hive-jdbc</artifactId>
            <version>0.12.0</version>
        </dependency>
        <dependency>
            <groupId>log4j</groupId>
            <artifactId>log4j</artifactId>
            <version>1.2.14</version>
        </dependency>
        <dependency>
            <groupId>com.fasterxml.jackson.core</groupId>
            <artifactId>jackson-core</artifactId>
            <version>2.5.3</version>
        </dependency>
        <dependency>
            <groupId>com.fasterxml.jackson.core</groupId>
            <artifactId>jackson-annotations</artifactId>
            <version>2.5.3</version>
        </dependency>
        <dependency>
            <groupId>org.codehaus.jackson</groupId>
            <artifactId>jackson-mapper-asl</artifactId>
            <version>1.9.0</version>
        </dependency>
        <dependency>
            <groupId>org.apache.spark</groupId>
            <artifactId>spark-streaming-kafka_2.10</artifactId>
            <version>${spark.version}</version>
        </dependency>
        <dependency>
            <groupId>mysql</groupId>
```

```
            <artifactId>mysql-connector-java</artifactId>
            <version>5.1.34</version>
        </dependency>
    </dependencies>
    <build>
        <plugins>
            <plugin>
                <artifactId>maven-assembly-plugin</artifactId>
                <version>2.3</version>
                <configuration>
                    <classifier>dist</classifier>
                    <appendAssemblyId>true</appendAssemblyId>
                    <descriptorRefs>
                        <descriptor>jar-with-dependencies</descriptor>
                    </descriptorRefs>
                </configuration>
                <executions>
                    <execution>
                        <id>make-assembly</id>
                        <phase>package</phase>
                        <goals>
                            <goal>single</goal>
                        </goals>
                    </execution>
                </executions>
            </plugin>

            <plugin>
                <artifactId>maven-compiler-plugin</artifactId>
                <configuration>
                    <source>1.7</source>
                    <target>1.7</target>
                </configuration>
            </plugin>

            <plugin>
                <groupId>net.alchim31.maven</groupId>
                <artifactId>scala-maven-plugin</artifactId>
                <version>3.2.2</version>
                <executions>
                    <execution>
                        <id>scala-compile-first</id>
                        <phase>process-resources</phase>
                        <goals>
                            <goal>compile</goal>
                        </goals>
```

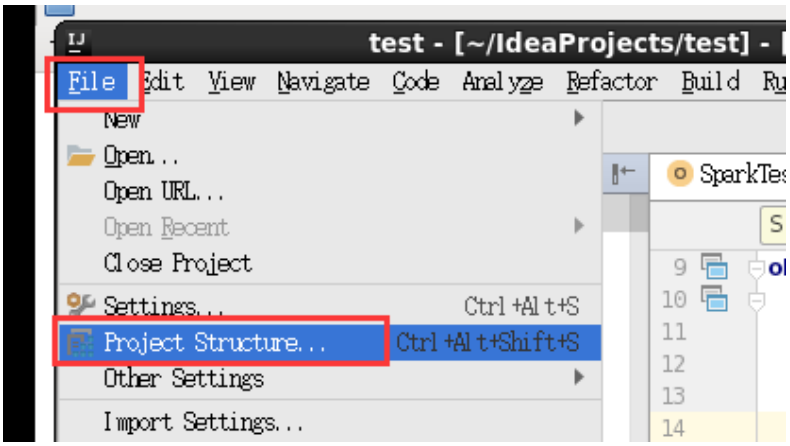
```
                    </execution>
                </executions>
                <configuration>
                    <scalaVersion>${scala.version}</scalaVersion>
                    <recompileMode>incremental</recompileMode>
                    <useZincServer>true</useZincServer>
                    <args>
                        <arg>-unchecked</arg>
                        <arg>-deprecation</arg>
                        <arg>-feature</arg>
                    </args>
                    <jvmArgs>
                        <jvmArg>-Xms1024m</jvmArg>
                        <jvmArg>-Xmx1024m</jvmArg>
                    </jvmArgs>
                    <javacArgs>
                        <javacArg>-source</javacArg>
                        <javacArg>${java.version}</javacArg>
                        <javacArg>-target</javacArg>
                        <javacArg>${java.version}</javacArg>
                        <javacArg>-Xlint:all,-serial,-path</javacArg>
                    </javacArgs>
                </configuration>
            </plugin>

        </plugins>
    </build>
</project>

```

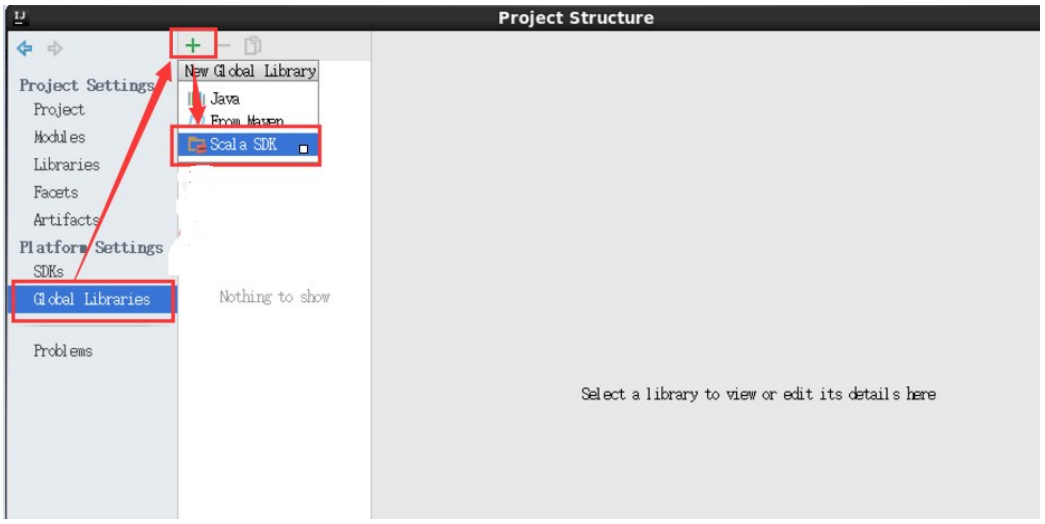
在项目上右键， 选择Maven-》Reimport，idea会自动从资源库下载需要的依赖jar包：整个过程可能需要很久（几个小时或者更久）。



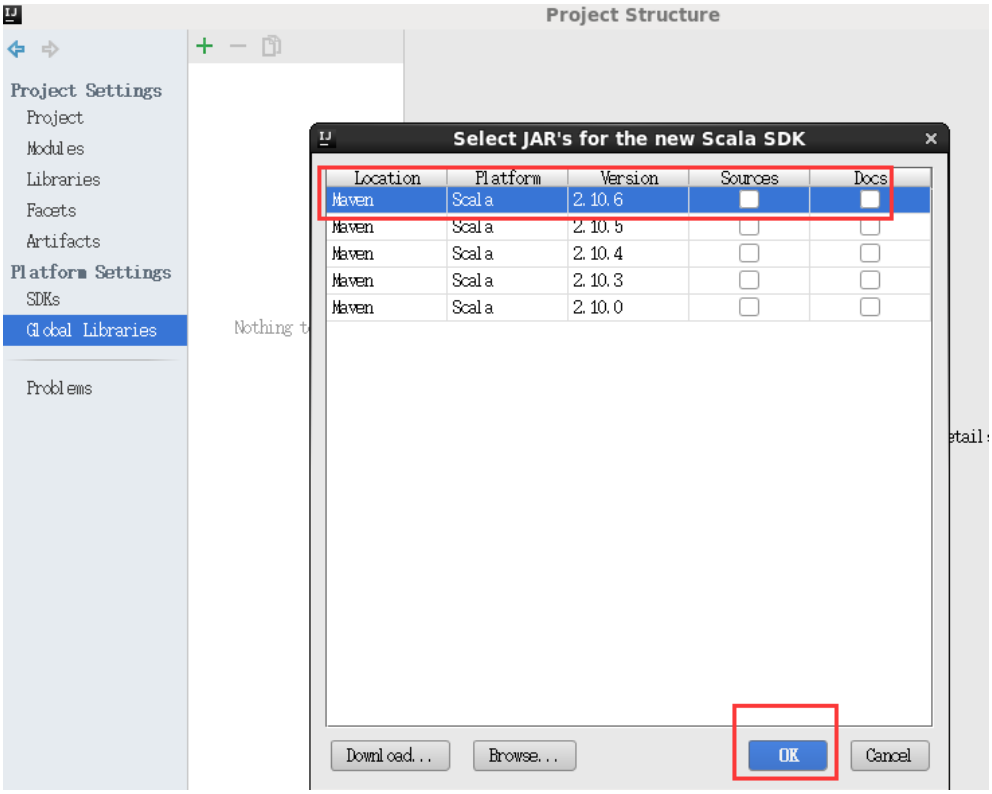
（6）增加项目的Scala支持

pom文件的依赖包下载完毕后， 增加项目的Scala支持。





选择Scala 2.10.6：



（7）创建项目目录结构

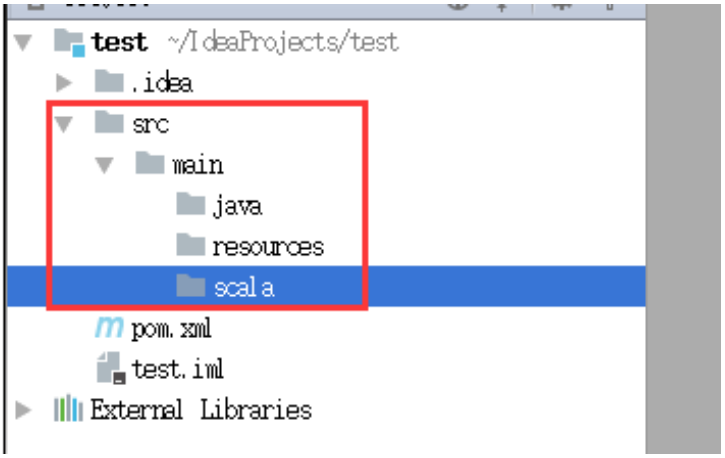
在项目下面， 创建如下层次的目录：

----------test/src/main

----------| java

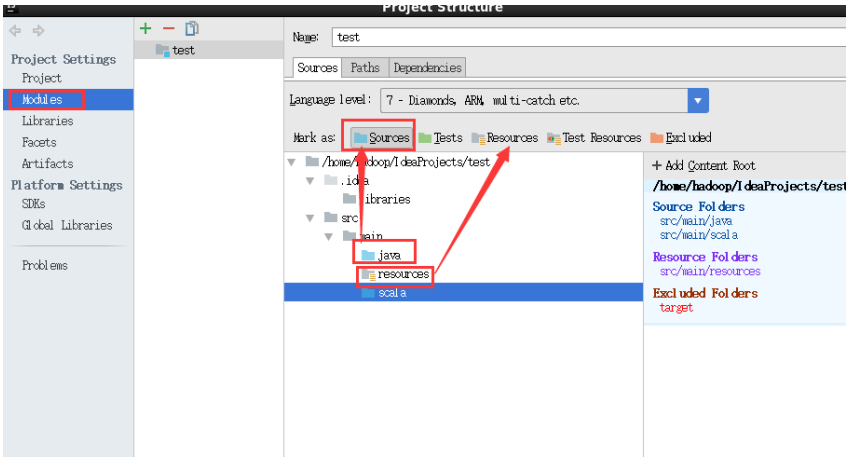
----------| resources

----------| scala



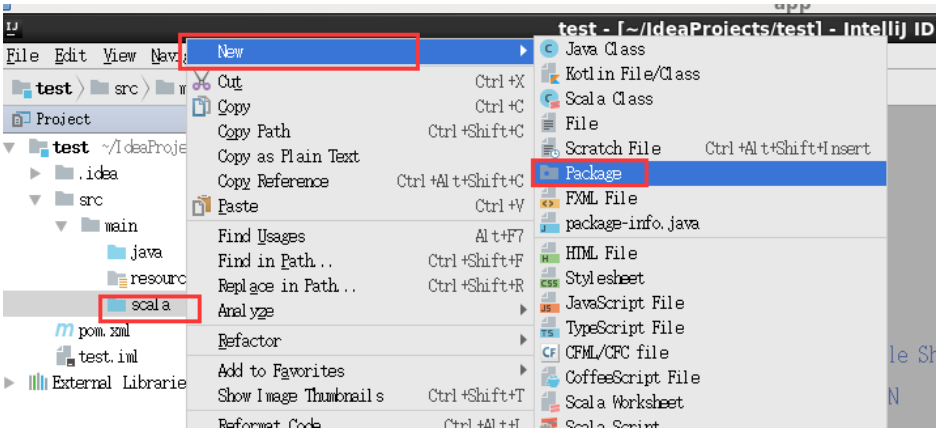
设置每个目录的模块功能：

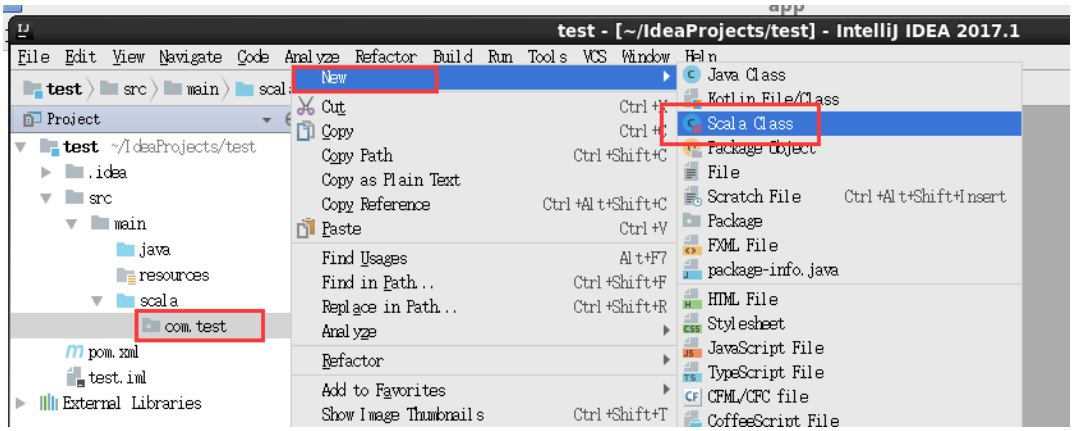
选择java， 设置为Sources。 选择resources目录， 设置为Resources。 如下：

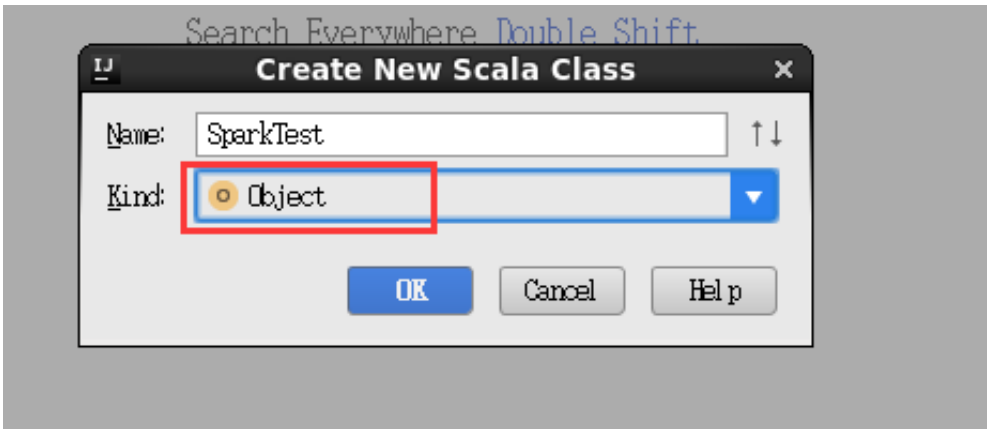


（8）创建scala测试程序

先创建包：



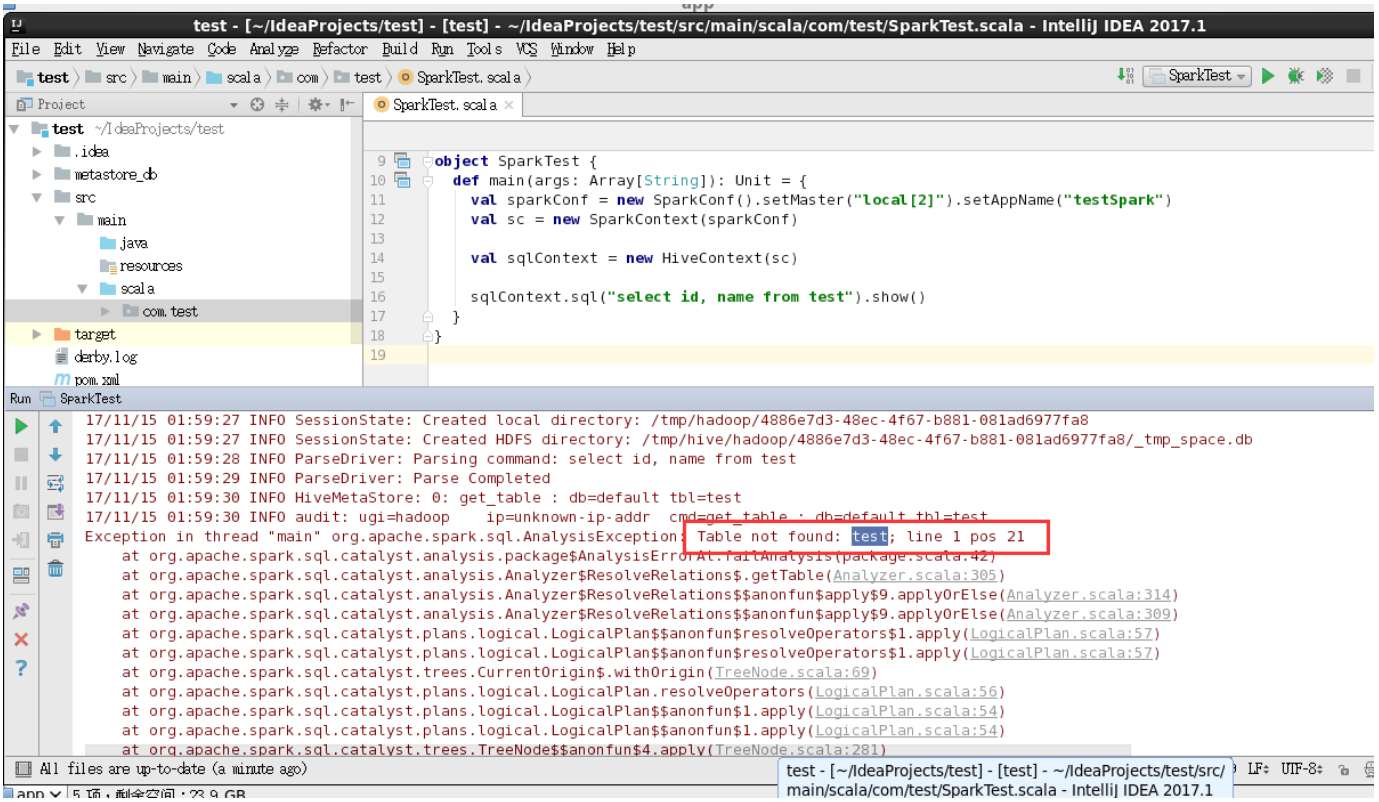




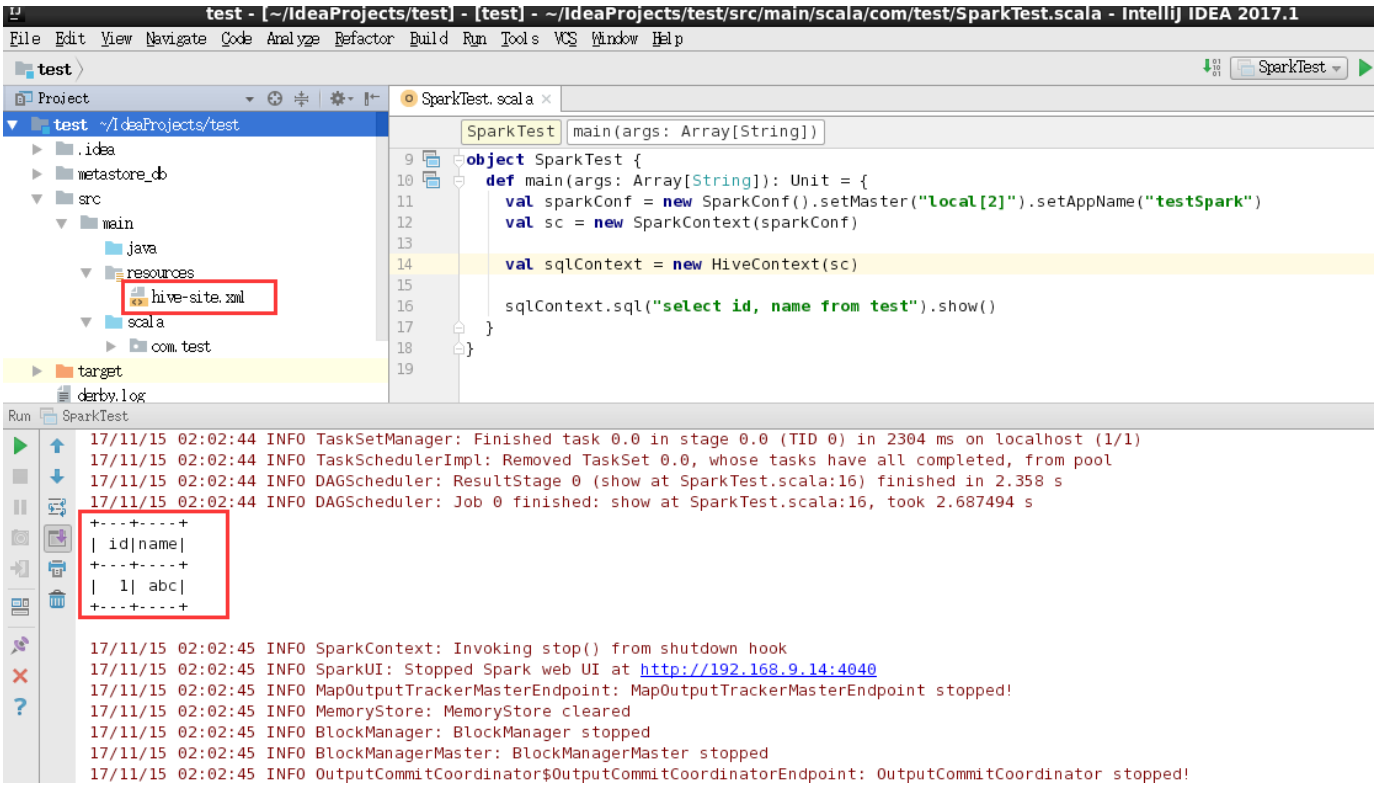
测试代码：

```
1  package com.test
2
3  import org.apache.spark.sql.hive.HiveContext
4  import org.apache.spark.{SparkConf, SparkContext}
5
6  /**
7    * Created by hadoop on 17-11-15.
8    */
9  object SparkTest {
10   def main(args: Array[String]): Unit = {
11     val sparkConf = new SparkConf().setMaster("local[2]").setAppName("testSpark")
12     val sc = new SparkContext(sparkConf)
13
14     val sqlContext = new HiveContext(sc)
15
16     sqlContext.sql("select id, name from test").show()
17   }
18 }
19
```

（9）读取hive表提示不存在的问题解决

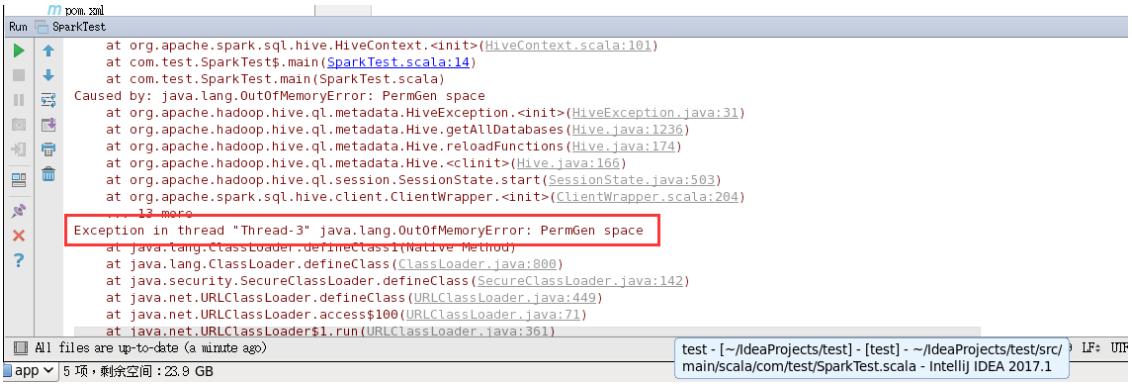


hive有test表， 但是提示test表不存在， 将hive-site.xml复制到resources目录即可。

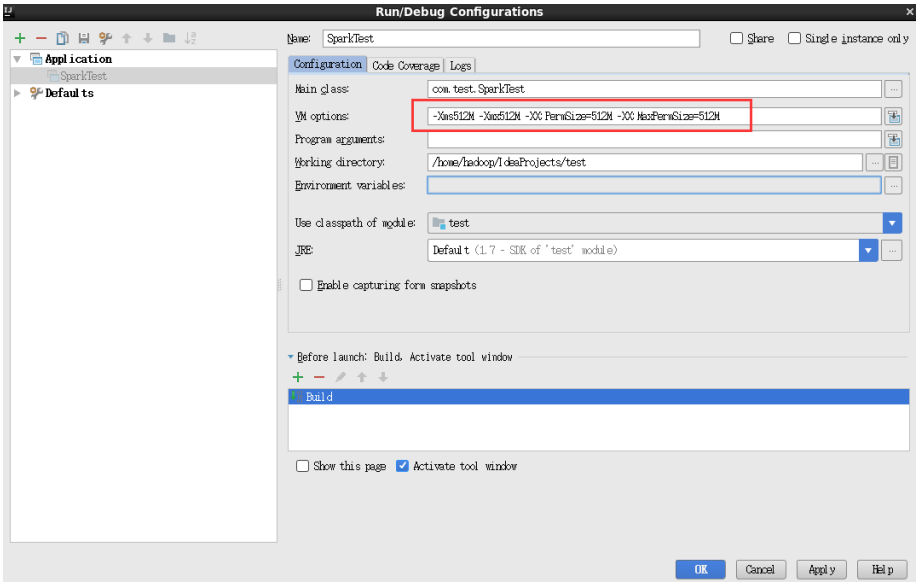


如果在idea操作hdfs， 需要将hadoop的core-site.xml复制到resources目录。

（9）如果报内存溢出错误，需要修改下IDEA的配置：



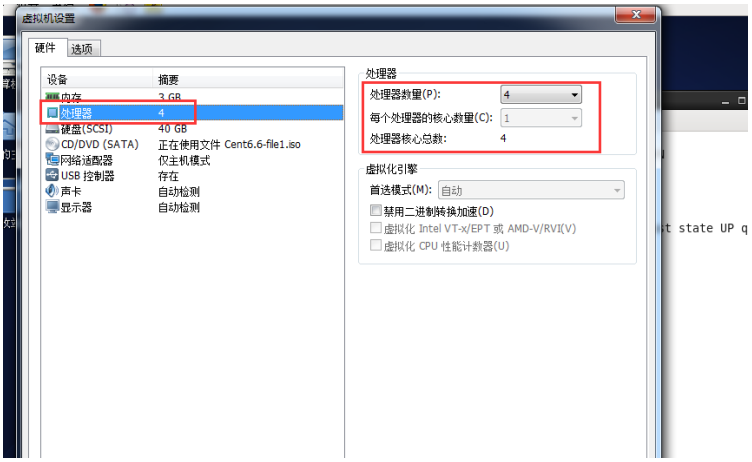
在VM options设置内存：-Xms512M -Xmx512M -XX:PermSize=512M -XX:MaxPermSize=512M



4. 其他

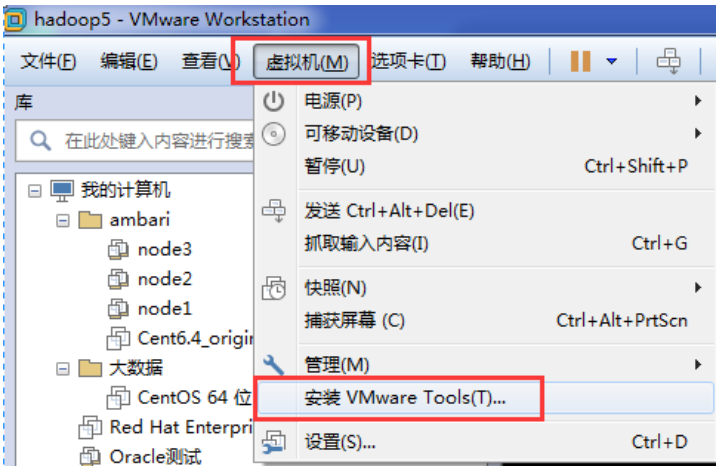
4.1 设置较大的虚拟机内存和CPU

内存和CPU尽量多分点，内存过小会导致spark作业提交失败。
CPU核心数太小会导致两个线程提交spark作业无法运行。

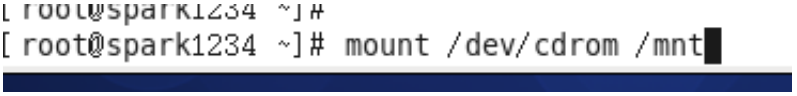


4.2 虚拟机和物理机无缝相互复制/粘贴

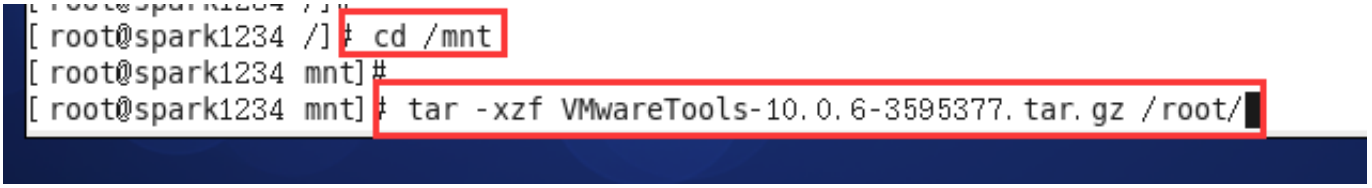
（1）选择安装VMware Tools



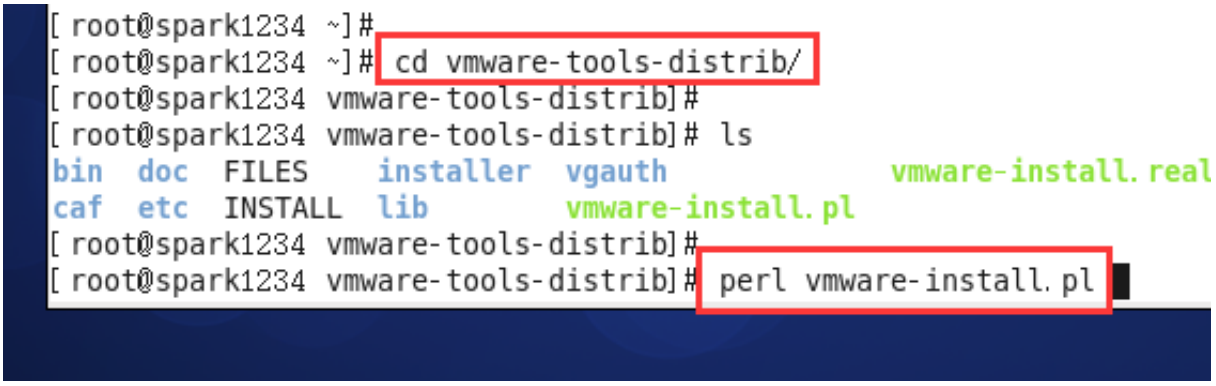
（2）挂载光盘



（3）进入挂载目录， 解压到/root/目录



（4）进入vmware-tools的解压目录， 一路回车即可：



安装完成后， 重启虚拟机， 物理机和虚拟机可以无缝复制粘贴。