# 大数据

## linux系统 之centos7 分支

大数据准备  三台电脑（虚拟机）

|  |  |  |  |
| --- | --- | --- | --- |
| 主机名 | Ip地址 | root账号 | 备注 |
| **node1** | **192.168.72.11** | **12345678** | 先集中力量配置好这台 |
| node2 | 192.168.72.12 | 12345678 | 等node1配置好，再克隆出来 |
| node3 | 192.168.72.13 | 12345678 | 等node1配置好，再克隆出来 |

### 1、配置信息

node1示例



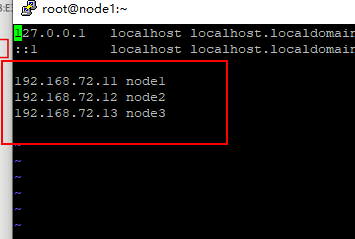
### 2、配置root账号的无密码登陆（可选）

使用无密码登陆 ssh node1 成功即可。方便我们登陆

### 3、绑定ip与主机名

修改文件/etc/hosts（ vi /etc/hosts ）入以下关联

192.168.72.11 node1  
192.168.72.12 node2  
192.168.72.13 node3



### 4、防火墙关闭——生产环境不要这样，非常不安全

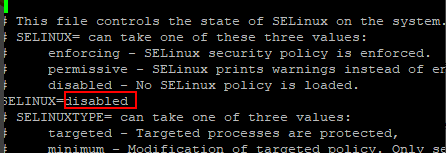
关： systemctl stop firewalld.service  
开机不启动： systemctl disable firewalld.service



### 5、关闭写入安全检查

修改文件 /etc/selinux/config （ vi /etc/selinux/config ）

将些文件的SELINUX属性，改成如下值：SELINUX=disabled



### 6、创建组与用户

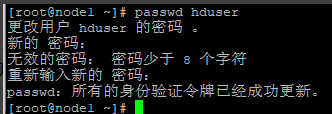
创建组，组名是 hadoop： groupadd hadoop

创建用户 并加入hadoop组： useradd -g hadoop hduser

修改hduser密码： passwd hduser

设置密码： 1234567

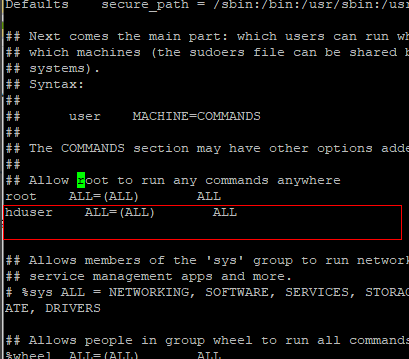
备注：当密码过于简单时，会有相关的安全提示，可以不理会，输入两次后，成功修改密码



### 7、给用户设置权限

附root级的权限给hduser用户

1. 修改文件为可修改状态 chmod 777 /etc/sudoers
2. 执行文件修改 vi /etc/sudoers



提示：在vi命令状态，输入 /root 可以快速定位此处。

1. 还原文件状态(必须还原只读状态) chmod 440 /etc/sudoers
2. 重启下系统 reboot

### 8、辅助——ntp时间同步组件使用（可选）

软件安装：

安装：sudo yum install ntp ntpdate -y

启动：sudo systemctl start ntpd

自启：sudo systemctl enable ntpd

设置：

采用主从方式。

我们使node1当作主。node2,node3当作从

#### 8.1、node1的配置（主）

sudo vi /etc/ntp.conf

设置自己的ip为主机： 192.168.72.11

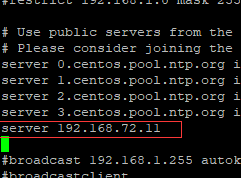
设置自己接受的客户端的ip范围： 192.168.72.0



#### 8.2、node2、node3的配置（从）

sudo vi /etc/ntp.conf

设置 时间服务器主机ip 192.168.72.11



#### 8.3、更新时间

所有节点启动时间同步  sudo timedatectl set-ntp yes

手动-立刻更新时间 sudo ntpdate -u 时间服务器 示例：sudo ntpdate -u 192.168.72.11

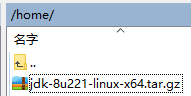
### 9、辅助——常用命令

|  |  |  |
| --- | --- | --- |
| 标题 | 命令 | 备注 |
| 重启命令 | reboot |  |
| 使用命令关机 | shutdown |  |
| **修改主机名** | hostnamectl set-hostname  新的主机名 |  |
| 查询ip相关 | ip addr |  |
| 查看防火墙的命令 | firewall-cmd --state |  |
|  |  |  |
|  |  |  |

## 配置jdk1.8

### 1、上传我们jdk到linux系统。

我的jdk文件名  jdk-8u221-linux-x64.tar.gz 。假设我们上传在 /home目录下



### 2、创建一个目录

mkdir /usr/java

### 3、解压jdk到/usr/java目录下

tar -zxvf /home/jdk-7u80-linux-x64.tar.gz  -C/usr/java

### 4、重命名我们的jdk文件名

cd /usr/java

mv jdk1.7.0\_80 jdk

经过这两步，我们得到jdk的根目录为 /usr/java/jdk

### 5、加入环境变量

编辑此文件/etc/profile( vi /etc/profile )

在文件最后，加入如下代码 :

export JAVA\_HOME=/usr/java/jdk  
export CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar:$CLASSPATH  
export PATH=$JAVA\_HOME/bin:$PATH

编辑好后，我们用source命令刷新环境变量

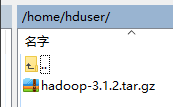
source /etc/profile

## hadoop 3.1.2版本

### 1、准备阶段(hduser)

使用hduser账号登陆

上传文件。



解压文件： tar -zxvf ~/hadoop-3.1.2.tar.gz

修改文件名为hadoop： mv hadoop-3.1.2 hadoop



备注：当我们不知道解压后的文件名是什么时  
我们可以使用 ls  命令列出当前目录的文件名或文件夹名

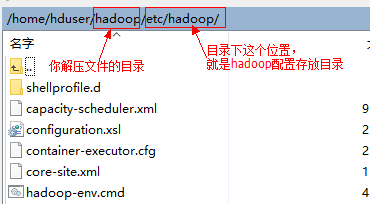
### 2、配置hadoop环境变量(hduser)

编译 sudo vi /etc/profile 文件，在最后加入以下内容：

export HADOOP\_HOME=/home/hduser/hadoop  
export PATH=$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin:$PATH  
export JAVA\_LIBRARY\_PATH=/home/hduser/hadoop/lib/native

### 3、集群配置(hduser)

#### 3.0、进入配置存放目录



#### 3.1、core-site.xml  配置

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

<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://node1:9000</value>

</property>

<property>

<name>io.file.buffer.size</name>

<value>131072</value>

</property>

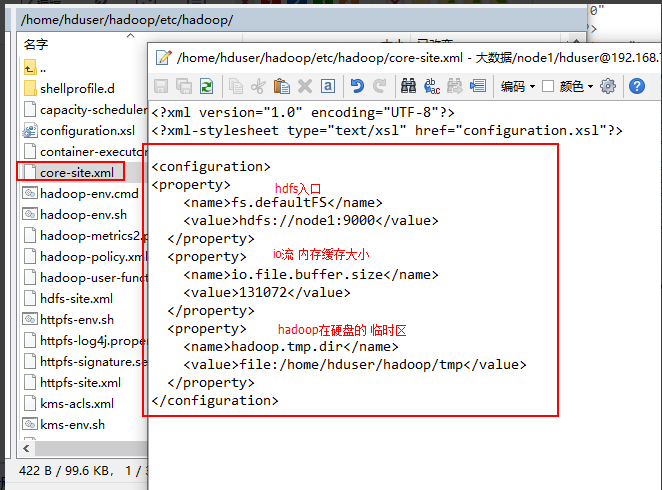
<property>

<name>hadoop.tmp.dir</name>

<value>file:/home/hduser/hadoop/tmp</value>

</property>

</configuration>

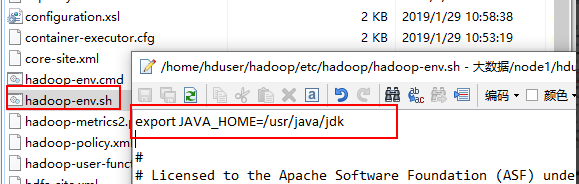


#### 3.2、hadoop-env.sh配置

在最顶上，

多加一行代码

export JAVA\_HOME=/usr/java/jdk



#### 3.3、hdfs-site.xml 配置

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

<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:/home/hduser/hadoop/dfs/data</value>

</property>

<property>

<name>dfs.blocksize</name>

<value>268435456</value>

</property>

<property>

<name>dfs.namenode.handler.count</name>

<value>100</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:/home/hduser/hadoop/dfs/data</value>

</property>

<property>

<name>dfs.namenode.secondary.http-address</name>

<value>node1:50090</value>

</property>

<property>

<name>dfs.webhdfs.enabled</name>

<value>true</value>

</property>

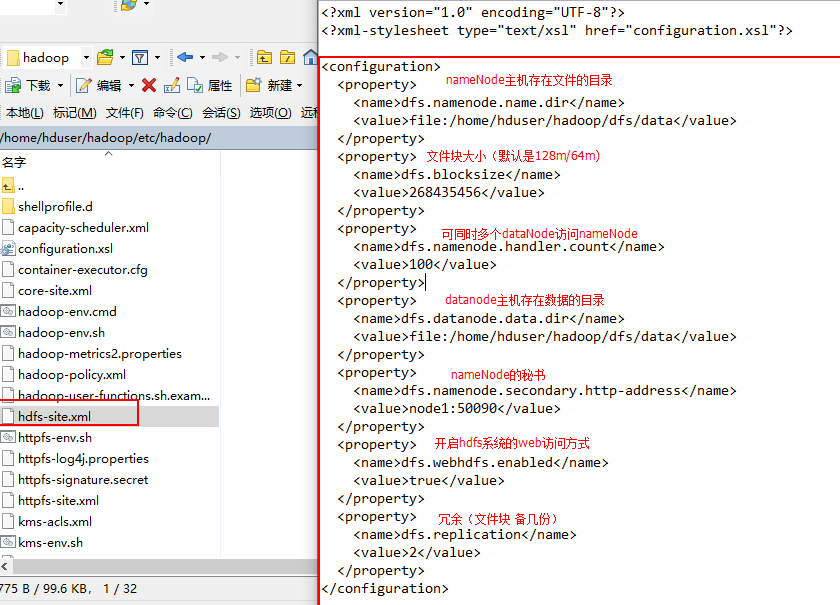
<property>

<name>dfs.replication</name>

<value>2</value>

</property>

</configuration>



#### 3.4、mapred-site.xml 配置

<?xml version="1.0"?>

<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

<property>

<name>mapreduce.jobhistory.address</name>

<value>node1:10020</value>

</property>

<property>

<name>mapreduce.jobhistory.webapp.address</name>

<value>node1:19888</value>

</property>

<property>

<name>mapreduce.map.memory.mb</name>

<value>4096</value>

</property>

<property>

<name>mapreduce.reduce.memory.mb</name>

<value>8192</value>

</property>

<property>

<name>mapreduce.map.java.opts</name>

<value>-Xmx3072m</value>

</property>

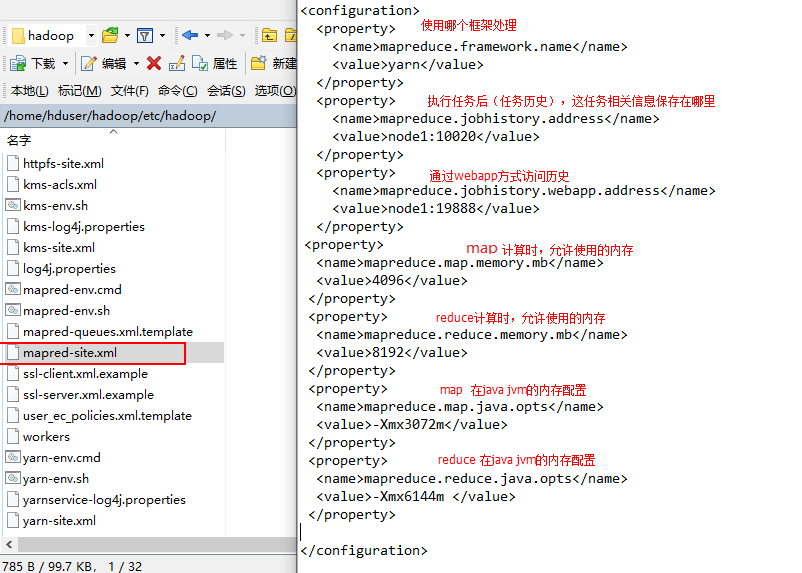
<property>

<name>mapreduce.reduce.java.opts</name>

<value>-Xmx6144m </value>

</property>

</configuration>



#### 3.5、yarn-site.xml 配置

<?xml version="1.0"?>

<configuration>

<property>

<name>yarn.resourcemanager.hostname</name>

<value>node1</value>

</property>

<property>

<name>yarn.resourcemanager.webapp.address</name>

<value>node1:8088</value>

</property>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

<property>

<name>yarn.resourcemanager.address</name>

<value>node1:8032</value>

</property>

<property>

<name>yarn.resourcemanager.admin.address</name>

<value>node1:8033</value>

</property>

<property>

<name>yarn.resourcemanager.scheduler.address</name>

<value>node1:8030</value>

</property>

<property>

<name>yarn.resourcemanager.resource-tracker.address</name>

<value>node1:8035</value>

</property>

<property>

<name>yarn.application.classpath</name>

<value>

/hadoop/hduser/etc/hadoop,

/home/hduser/hadoop/share/hadoop/common/\*,

/home/hduser/hadoop/share/hadoop/common/lib/\*,

/home/hduser/hadoop/share/hadoop/mapreduce/\*,

/home/hduser/hadoop/share/hadoop/mapreduce/lib/\*,

/home/hduser/hadoop/share/hadoop/hdfs/\*,

/home/hduser/hadoop/share/hadoop/hdfs/lib/\*,

/home/hduser/hadoop/share/hadoop/yarn/\*,

/home/hduser/hadoop/share/hadoop/yarn/lib/\*

</value>

</property>

<property>

<name>mapreduce.application.classpath</name>

<value>

/hadoop/hduser/etc/hadoop,

/home/hduser/hadoop/share/hadoop/common/\*,

/home/hduser/hadoop/share/hadoop/common/lib/\*,

/home/hduser/hadoop/share/hadoop/mapreduce/\*,

/home/hduser/hadoop/share/hadoop/mapreduce/lib/\*,

/home/hduser/hadoop/share/hadoop/hdfs/\*,

/home/hduser/hadoop/share/hadoop/hdfs/lib/\*,

/home/hduser/hadoop/share/hadoop/yarn/\*,

/home/hduser/hadoop/share/hadoop/yarn/lib/\*

</value>

</property>

</configuration>

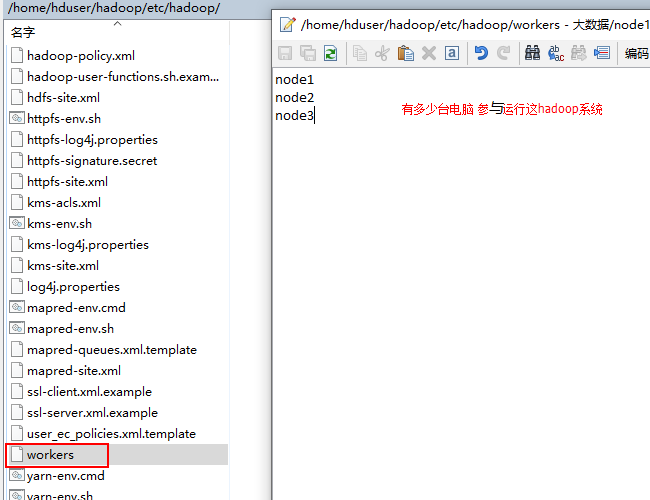
#### 3.6、yarn-env.sh配置

在最顶上，

多加一行代码

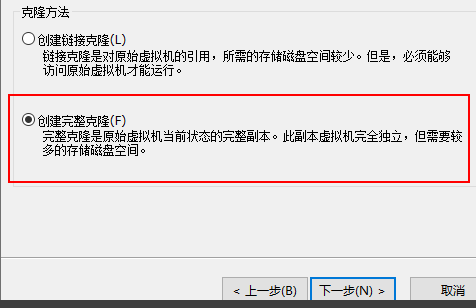
export JAVA\_HOME=/usr/java/jdk

#### 3.7、workers 配置



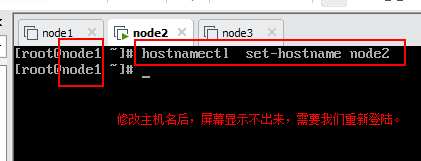
### 4、关要克隆2台(先把node1关机)

采用完整克隆方式，把node1克隆两次。分别为node2、node3

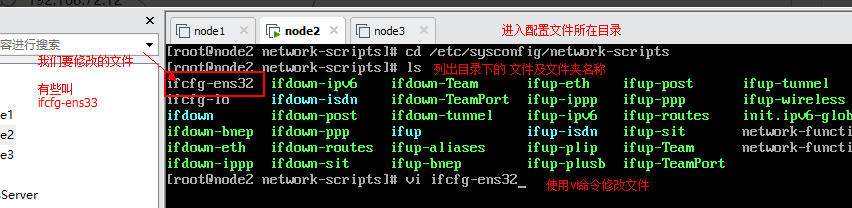


#### 4.1、针对克隆机node2

**修改主机名**

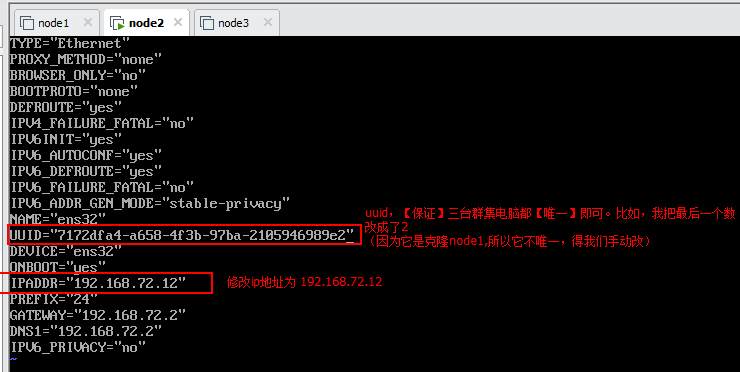


如图，修改文件



修改ip与uui

备注：uuid类似人类的身份证，表示唯一的编号。因为我们复制node1，所以，需要把它改成唯一！注意长度与格式不能为



修改完后，重启一次。

#### 4.2、针对克隆机node3

修改主机名为 node3 命令参考node2

修改ip为192.168.72.13

修改 uui为唯一即可，不要与node1、node2的相同。唯一即可！

### 5、确保集群电脑 能无密码相互访问

#### 5.1、node1示例（hduser）

ssh-keygen -t rsa

ssh-copy-id node1

第一次会问你是否确认两台主机建立的信任关系，输入yes，再按回车键

第一次会要求你输入一次localhost(指本机）用户的密码

成功后，我们通过ssh node1可登录测试，再exit退出（如果一下就退出到root的登陆界面，ssh-copy-id node1不成功，或ssh-keygen -t rsa不成功）ssh node1的连接

ssh-copy-id node2 即：把node1的凭证传到node2中

第一次会问你是否确认两台主机建立的信任关系，输入yes，再按回车键

第一次会要求你输入一次node2(指另一台虚拟机上的用户hduser）用户的密码

成功后，我们通过ssh node2可登录测试，再exit退出（如果一下就退出到root的登陆界面，ssh-copy-id node2不成功）ssh node2的连接

ssh-copy-id node3 即：把node1的凭证传到node3中

第一次会问你是否确认两台主机建立的信任关系，输入yes，再按回车键

第一次会要求你输入一次node3(指另一台虚拟机上的用户hduser）用户的密码

成功后，我们通过ssh node3可登录测试，再exit退出（如果一下就退出到root的登陆界面，ssh-copy-id node3不成功）ssh node3的连接

#### 5.2、node2示例（hduser）

ssh-keygen -t rsa

ssh-copy-id node2

第一次会问你是否确认两台主机建立的信任关系，输入yes，再按回车键

第一次会要求你输入一次localhost(指本机）用户的密码

成功后，我们通过ssh node2可登录测试，再exit退出（如果一下就退出到root的登陆界面，ssh-copy-id node2不成功，或ssh-keygen -t rsa不成功）ssh localhost的连接

ssh-copy-id node1 即：把node1的凭证传到node1中

第一次会问你是否确认两台主机建立的信任关系，输入yes，再按回车键

第一次会要求你输入一次node1 (指另一台虚拟机上的用户hduser）用户的密码

成功后，我们通过ssh node1可登录测试，再exit退出（如果一下就退出到root的登陆界面，ssh-copy-id node1不成功）ssh node2的连接

ssh-copy-id node3 即：把node1的凭证传到node3中

第一次会问你是否确认两台主机建立的信任关系，输入yes，再按回车键

第一次会要求你输入一次node3(指另一台虚拟机上的用户hduser）用户的密码

成功后，我们通过ssh node3可登录测试，再exit退出（如果一下就退出到root的登陆界面，ssh-copy-id node3不成功）ssh node3的连接

#### 5.3、node3示例（hduser）

ssh-keygen -t rsa

ssh-copy-id node3

第一次会问你是否确认两台主机建立的信任关系，输入yes，再按回车键

第一次会要求你输入一次localhost(指本机）用户的密码

成功后，我们通过ssh node3可登录测试，再exit退出（如果一下就退出到root的登陆界面，ssh-copy-id node3不成功，或ssh-keygen -t rsa不成功）ssh node3的连接

ssh-copy-id node1 即：把node1的凭证传到node1中

第一次会问你是否确认两台主机建立的信任关系，输入yes，再按回车键

第一次会要求你输入一次node1 (指另一台虚拟机上的用户hduser）用户的密码

成功后，我们通过ssh node1可登录测试，再exit退出（如果一下就退出到root的登陆界面，ssh-copy-id node1 不成功）ssh node2的连接

ssh-copy-id node2 即：把node1的凭证传到node3中

第一次会问你是否确认两台主机建立的信任关系，输入yes，再按回车键

第一次会要求你输入一次node2 (指另一台虚拟机上的用户hduser）用户的密码

成功后，我们通过ssh node2可登录测试，再exit退出（如果一下就退出到root的登陆界面，ssh-copy-id node2不成功）ssh node2的连接

### 6、初始化与启动(hduser)

6.1、进入hadoop 目录: cd ~/hadoop

6.2、初始化（危险，一次就好）：bin/hdfs namenode -format

备注：多次初始化时，会导致主节点与数据节点的版本不一致。

如果不小心多次初始化，需要把数据节点生成的数据全部清空。  
再重新初始化

### 7、启动、停止、报告状态、角色

在目录下（ cd ~/hadoop ）

启动 sbin/start-all.sh

报告状态 bin/hdfs dfsadmin -report

停止 sbin/stop-all.sh

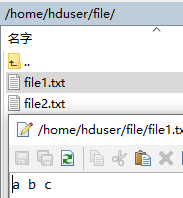
查询当前电脑扮演的角色：jps

### 8、测试运行jar程序（【node1】主机，hduser用户）

#### 8.1、准备

在/home/hduser目录下，创建一个文件夹file

再在/home/hduser/file里面放入两个文本(内容随意。最好是一些单词，单词间有空格)，分别取名叫file1.txt file2.txt (名字统一却内容不为空，主要方便一会我们练习用）。如：



#### 8.2、测试

确认我们的hadoop已经运行，无错。

使用终端进入hadoop目录（ cd ~/hadoop ）：

创建目录：bin/hadoop fs -mkdir /input2

推送文件：bin/hadoop fs -put ~/file/file\*.txt /input2/

列出文件：bin/hadoop fs -ls /input2/

执行我们的入门测试：

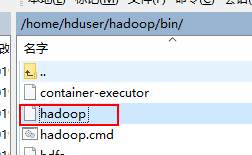
bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.1.2.jar wordcount /input2/ /output2/wordcount1

查看程序执行后的结果

bin/hadoop fs -cat /output2/wordcount1/\*

### 9、hdfs常用命令

执行命令的脚本： hadoop目录/bin/hadoop



语法：hadoop fs <args>

|  |  |  |  |
| --- | --- | --- | --- |
| **命令** | **说明** | **示例** | **备注** |
| hadoop fs -mkdir <paths> | 创建目录 | hadoop fs -mkdir /user | 在HDFS中创建“/user”目录 |
| hadoop fs -mkdir /user/hadoop/dir1 /user/hadoop/dir2 | 同时创建多个目录 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### 10、使用hdfs api——java编程方式

示例项目 https://github.com/qiyueW/hadoop.file.git

### 11、编写MapReduce程序

#### Map步骤

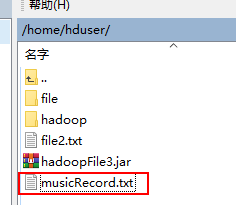
继承**public** **class** Mapper<KEYIN, VALUEIN, KEYOUT, VALUEOUT>

重写里面的map方法。

示例



### 12、执行MapReduce程序



把我们的数据，放入HDFS中

先创建一个music目录（在hdfs中）：hadoop fs -mkdir /music

把linux系统中的文件，存入hdfs中：hadoop fs -put ~/musicRecord.txt /music

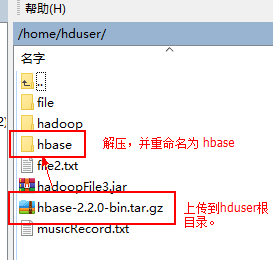
查看目录是否有你存入的文件（可选）：hadoop fs -ls /music

执行：hadoop jar ~/hadoopFile3.jar /music/\* /output4

查看结果：hadoop fs -cat /output4/\*

## HBase 2.2.0版本

### 1、准备阶段（node1,hduser账号）

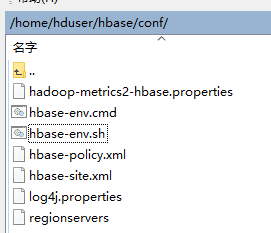


解压命令 tar -zxvf hbase-2.2.0-bin.tar.gz

重命名 mv hbase-2.2.0 hbase

### 2、配置（node1,hduser账号）

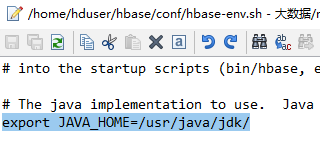
进入 /home/hduser/hbase/conf 目录



#### 2.1、配置hbase-env.sh

2.1.1在最顶部，追加这项代码： export JAVA\_HOME=/usr/java/jdk/

2.1.2打开配置（去掉选项前面的#号即可） export HBASE\_MANAGES\_ZK=true





#### 2.2、配置 hbase-site.xml

<?xml version="1.0"?>

<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>

<property>

<name>hbase.rootdir</name>

<value>hdfs://node1:9000/hbase</value>

</property>

<property>

<name>hbase.cluster.distributed</name>

<value>true</value>

</property>

<property>

<name>hbase.zookeeper.quorum</name>

<value>node1,node2,node3</value>

</property>

<property>

<name>hbase.zookeeper.property.dataDir</name>

<value>/home/hduser/hbase/zookeeper/data</value>

</property>

<property>

<name>hbase.unsafe.stream.capability.enforce</name>

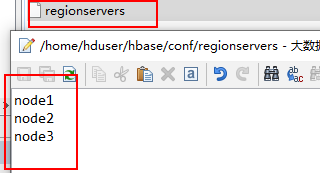
<value>false</value>

</property>

</configuration>

#### 2.3、配置regionservers

在文件里，加入 node1 node2 node3

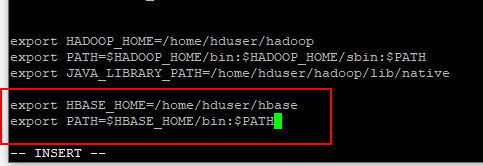


#### 2.4、设置环境变量（所有的主机node1,node2,node3都需要配置）

sudo vi /etc/profile

然后在最后追加如下内容,然后保存退出。

export HBASE\_HOME=/home/hduser/hbase  
export PATH=$HBASE\_HOME/bin:$PATH



#### 2.5、解决冲突（所有的主机node1,node2,node3都需要配置）

**hadoop与hbase有一个jar冲突**

2.5.1 删除/home/hduser/hadoop/share/hadoop/common/lib下的**slf4j-log4j12-1.7.25.jar**包

删除

2.5.2 把hbase的jar包共享出去

**sudo vi /etc/profile**

**在文件最后加入如下代码：**

export CLASSPATH=/home/hduser/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.25.jar:$CLASSPATH



注意：共享出去后，以后，所有引用此jar的项目，都不用再添加此jar包了！否则会报冲突

#### 2.6、执行一下命令

让之前的环境变量配置生效

source /etc/profile

### 3、同步node1配置到node2、node3主机

#### 3.1 同步hbase配置(在node1 hduser账号的控制台下执行)

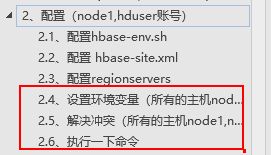
复制到node2主机： scp -r /home/hduser/hbase/  hduser@node2:/home/hduser/

复制到node3主机： scp -r /home/hduser/hbase/  hduser@node3:/home/hduser/

#### 3.2 单独配置（都是hduser账号）

同步完后，要登陆node2、node3，进行配置

**参考【配置】第2.4、2.5、2.6项的操作**



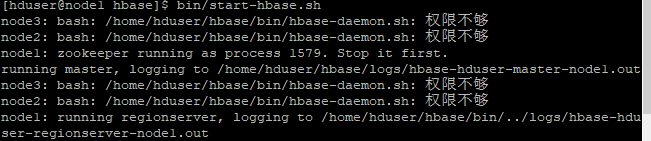
### 4、启动

4.1 确保hadoop的hdfs系统已经启动

4.2 进入hbase目录： cd ~/hbase

4.3  启动hbase：bin/start-hbase.sh

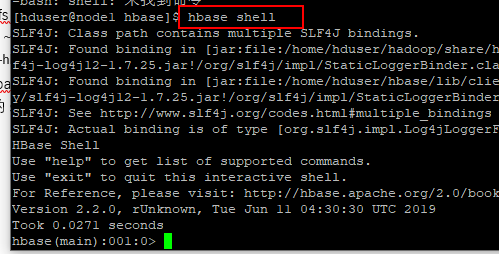
启动报权限不够



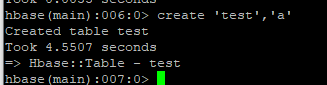
再从机配置 : chmod -R 777 ./

4.4 测试是否成功： hbase shell

如果能进入到 bhase 的shell控制台（shell 接收指令的窗口）



测试一下创建表的命令：create 'test','a'



**退出shell界面，输入exit即可**



### 5、通过shell操作hbase数据库

在hbase目录下（cd ~/hbase ）

执行命令： hbase shell 进入shell界面

5.1 使用shell命令创建一个表

**create 'student','info','lp'**