# HBase完全分布式集群安装

HBase集群建立在hadoop集群基础之上，所以在搭建HBase集群之前需要把Hadoop集群搭建起来

## 名词解释

### Hadoop

[Hadoop](https://baike.baidu.com/item/Hadoop" \t "https://baike.baidu.com/item/Hadoop/_blank)是一个由Apache基金会所开发的[分布式系统](https://baike.baidu.com/item/%E5%88%86%E5%B8%83%E5%BC%8F%E7%B3%BB%E7%BB%9F/4905336" \t "https://baike.baidu.com/item/Hadoop/_blank)基础架构。

用户可以在不了解分布式底层细节的情况下，开发分布式程序。充分利用集群的威力进行高速运算和存储。

Hadoop实现了一个[分布式文件系统](https://baike.baidu.com/item/%E5%88%86%E5%B8%83%E5%BC%8F%E6%96%87%E4%BB%B6%E7%B3%BB%E7%BB%9F/1250388" \t "https://baike.baidu.com/item/Hadoop/_blank)（Hadoop Distributed File System），简称HDFS。HDFS有高[容错性](https://baike.baidu.com/item/%E5%AE%B9%E9%94%99%E6%80%A7/9131391" \t "https://baike.baidu.com/item/Hadoop/_blank)的特点，并且设计用来部署在低廉的（low-cost）硬件上；而且它提供高吞吐量（high throughput）来访问[应用程序](https://baike.baidu.com/item/%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F/5985445" \t "https://baike.baidu.com/item/Hadoop/_blank)的数据，适合那些有着超大数据集（large data set）的应用程序。HDFS放宽了（relax）POSIX的要求，可以以流的形式访问（streaming access）文件系统中的数据。

Hadoop的框架最核心的设计就是：HDFS和MapReduce。HDFS为海量的数据提供了存储，而MapReduce则为海量的数据提供了计算。

### HBase

HBase是一个分布式的、面向列的开源数据库，该技术来源于 Fay Chang 所撰写的Google论文“Bigtable：一个结构化数据的[分布式存储系统](https://baike.baidu.com/item/%E5%88%86%E5%B8%83%E5%BC%8F%E5%AD%98%E5%82%A8%E7%B3%BB%E7%BB%9F/6608875" \t "https://baike.baidu.com/item/_blank)”。就像Bigtable利用了Google文件系统（File System）所提供的分布式数据存储一样，HBase在Hadoop之上提供了类似于Bigtable的能力。HBase是Apache的Hadoop项目的子项目。HBase不同于一般的关系数据库，它是一个适合于非结构化数据存储的数据库。另一个不同的是HBase基于列的而不是基于行的模式。

### zookeeper

ZooKeeper是一个[分布式](https://baike.baidu.com/item/%E5%88%86%E5%B8%83%E5%BC%8F/19276232" \t "https://baike.baidu.com/item/_blank)的，开放源码的[分布式应用程序](https://baike.baidu.com/item/%E5%88%86%E5%B8%83%E5%BC%8F%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F/9854429" \t "https://baike.baidu.com/item/_blank)协调服务，是[Google](https://baike.baidu.com/item/Google" \t "https://baike.baidu.com/item/_blank)的Chubby一个[开源](https://baike.baidu.com/item/%E5%BC%80%E6%BA%90/246339" \t "https://baike.baidu.com/item/_blank)的实现，是Hadoop和Hbase的重要组件。它是一个为分布式应用提供一致性服务的软件，提供的功能包括：配置维护、域名服务、分布式同步、组服务等。

ZooKeeper的目标就是封装好复杂易出错的关键服务，将简单易用的接口和性能高效、功能稳定的系统提供给用户。

准备工作：

Jdk,Hadoop,hbase

4台服务器

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 服务器 | zk | Hadoop  (nameNode) | Hadoop  (datanode) | Hbase  (master) | Hbase  (regionserver) |
| Node1  (192.168.12.10)  192.168.11.234 | × | √ | × | √(主) | × |
| Node2  (192.168.13.39)  192.168.11.235 | √ | × | √ | √(备) | × |
| Node3  (192.168.13.40)  192.168.11.236 | √ | × | √ | × | √(从) |
| Node4  (192.168.13.41)  192.168.11.239 | √ | × | √ | × | √(从) |
| (192.168.12.11)  192.168.11.233 |  |  |  |  |  |

## 安装JDK

1）将jdk安装包放在家目录下

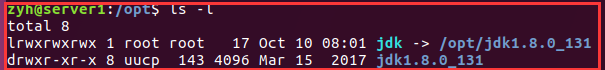
2）解压到opt目录下

|  |
| --- |
| #sudo tar zxvf jdk-8u191-linux-x64.tar.gz -C /opt/ |

此时在/opt目录下：会有一个jdk1.8.0\_131

3）创建软链接

|  |
| --- |
| #sudo ln -snf /opt/jdk1.8.0\_191/ /opt/jdk |



　　　　注意：创建软连接的目的是为了，我们在做项目的时候，可能会用到不同的jdk版本，这是要换软件的话，只需要修改软链接就可以了。而不用修改配置文件。

　　4）配置环境变量

　　　　局部环境变量：~/.bashrc  
　　　　全局环境变量：/etc/profile(编辑profile文件)

|  |
| --- |
| #vim /etc/profile |

添加如下

|  |
| --- |
| export JAVA\_HOME=/opt/jdk export JRE\_HOME=$JAVA\_HOME/jre export CLASSPATH=.:$JAVA\_HOME/lib  export PATH=$PATH:$JAVA\_HOME/bin |

source /etc/profile（更新配置文件）

　　5）查看是否安装成功

　　　java、javac、java -version

## hadoop搭建分布式集群

准备工作：

关闭防火墙

关闭： systemctl stop firewalld

开机禁用 ： systemctl disable firewalld

查看状态： systemctl status firewalld

4台服务器都添加如下内容，内容一样

|  |
| --- |
| # vi /etc/hosts |

添加内容

|  |
| --- |
| 192.168.12.10 node1  192.168.13.39 node2  192.168.13.40 node3  192.168.13.41 node4  192.168.11.234 node1  192.168.11.235 node2  192.168.11.236 node3  192.168.11.239 node4 |

4台服务器都配置自己的节点

|  |
| --- |
| # vi /etc/sysconfig/network |

添加内容

|  |
| --- |
| NETWORKING=yes  HOSTNAME=node1 |

重启Linux

|  |
| --- |
| #reboot |

或者

|  |
| --- |
| #hostname node1 (立马生效) |

### 安装hadoop

下载包

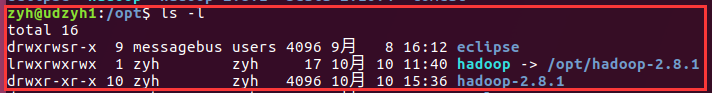
如：cenos-6.5-hadoop-2.6.4.tar.gz

（1）解压hadoop安装包到opt目录下

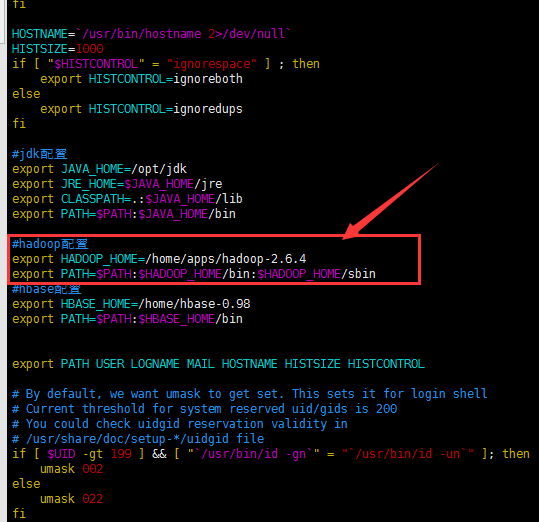
|  |
| --- |
| #sudo  tar zxvf cenos-6.5-hadoop-2.6.4.tar.gz -C /home |

（2）创建软链接

|  |
| --- |
| #ln -snf /home/hadoop-2.6.4/ /home/hadoop |

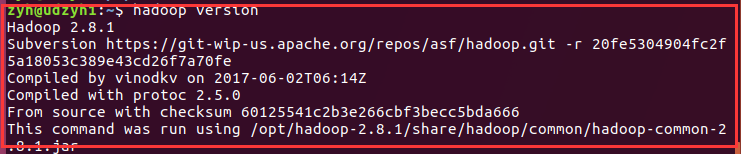


（3）配置环境变量

在/etc/profile文件中加入以下内容：  


|  |
| --- |
| #hadoop配置  export HADOOP\_HOME=/home/hadoop-2.6.4  export PATH=$PATH:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin |

source /etc/profile（刷新环境变量）

（4）使用hadoop version命令测试是否配置成功　

### 配置hadoop

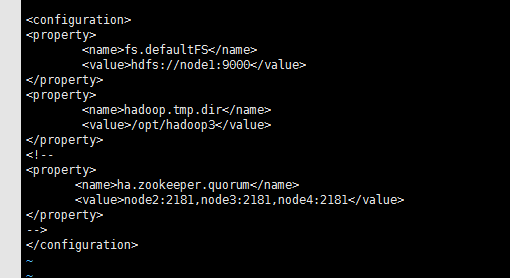
进入：/home/hadoop-2.6.4/etc/hadoop

#### 第一个：hadoop-env.sh

|  |
| --- |
| #vim hadoop-env.sh |

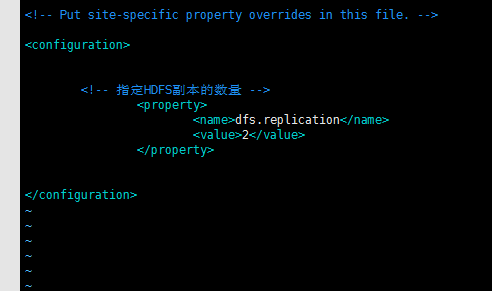
|  |
| --- |
| export JAVA\_HOME=/jdk/opt |

#### 第二个：core-site.xml



|  |
| --- |
| <configuration>  <!-- 指定HADOOP所使用的文件系统schema（URI）,HDFS的老大（NameNode）的地址 -->  <property>  <name>fs.defaultFS</name>  <value>hdfs://node1:9000</value>  </property>  <!-- 指定hadoop运行时产生文件的存储目录 -->  <property>  <name>hadoop.tmp.dir</name>  <value>/home/hadoop-dir</value>  </property>  </configuration> |

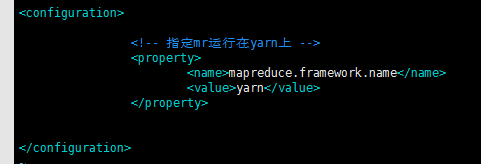
#### 第三个：hdfs-site.xml



|  |
| --- |
| <configuration>  <!-- 加上服务名，也是ok的 -->  <property>  <name>dfs.nameservices</name>  <value>bjsxt</value>  </property>  <!-- 指定HDFS副本的数量 -->  <property>  <name>dfs.replication</name>  <value>2</value>  </property>  </configuration> |

#### 第四个：mapred-site.xml

(mv mapred-site.xml.template mapred-site.xml)



|  |
| --- |
| <configuration>  <!-- 指定mr运行在yarn上 -->  <property>  <name>mapreduce.framework.name</name>  <value>yarn</value>  </property>  </configuration> |

#### 第五个：yarn-site.xml



|  |
| --- |
| <configuration>  <!-- Site specific YARN configuration properties -->  <!-- 指定YARN的老大（ResourceManager）的地址 -->  <property>  <name>yarn.resourcemanager.hostname</name>  <value>node1</value>  </property>  <!-- reducer获取数据的方式 -->  <property>  <name>yarn.nodemanager.aux-services</name>  <value>mapreduce\_shuffle</value>  </property>  </configuration> |

把第一个配置好的复制到其他几台服务器

|  |
| --- |
| #scp -r /home/apps root@node1:/home |

然后输入密码即可

hadoop namenode -format（只对第一台格式化） namenode

|  |
| --- |
| #hadoop namenode -format |

#### 最后启动

##### 单台启动方法

（第一台服务器）

|  |
| --- |
| #hadoop-daemon.sh start namenode |

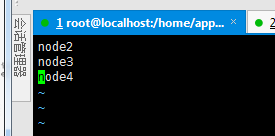
其他几台

|  |
| --- |
| #hadoop-daemon.sh start datanode |

##### 多台启动方法

编辑 slaves文件

|  |
| --- |
| #vim /home/hadoop-2.6.4/etc/hadoop/slaves |



|  |
| --- |
| node2  node3  node4 |

最后启动命令

|  |
| --- |
| #start-all.sh |

或者

|  |
| --- |
| 先启动HDFS  #start-dfs.sh  再启动YARN  #start-yarn.sh |

验证是否启动成功

|  |
| --- |
| #jps |

显示内容

|  |
| --- |
| 27408 NameNode  28218 Jps  27643 SecondaryNameNode  28066 NodeManager  27803 ResourceManager  27512 DataNode |

#### 停止方法

（第一台服务器）

|  |
| --- |
| #hadoop-daemon.sh stop namenode |

其他几台

|  |
| --- |
| #hadoop-daemon.sh stop datanode |

停止还可以用方法kill

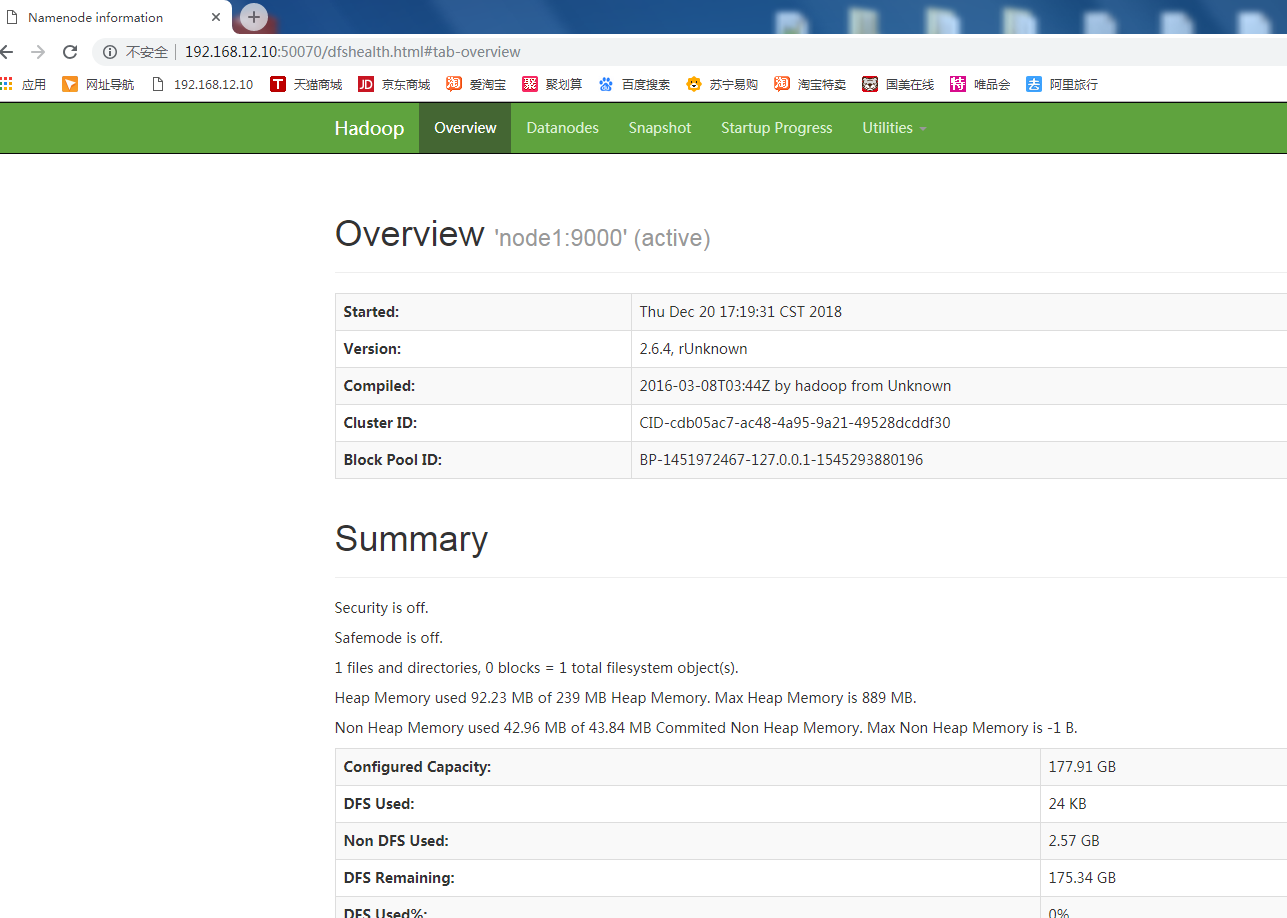
|  |
| --- |
| #Kill -9 xxx(用jps查看) |

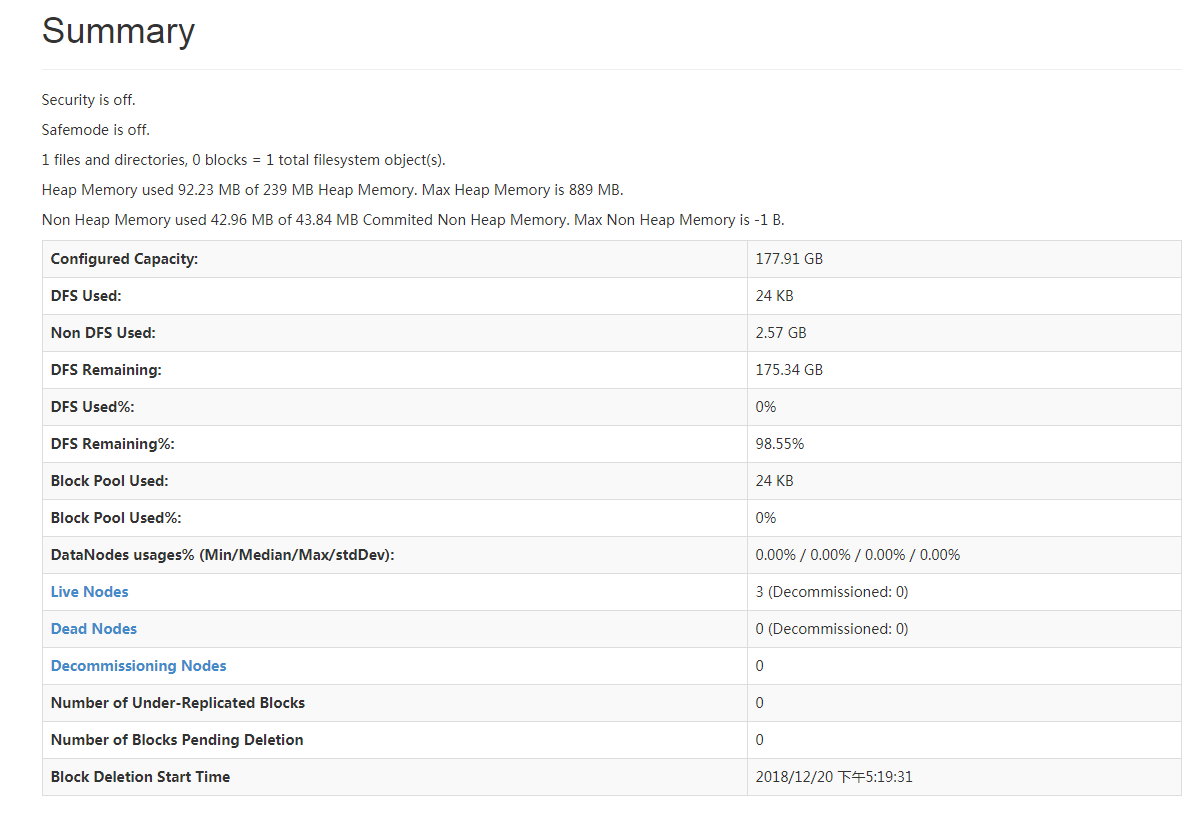
多台服务器一起停止

|  |
| --- |
| #stop-all.sh |

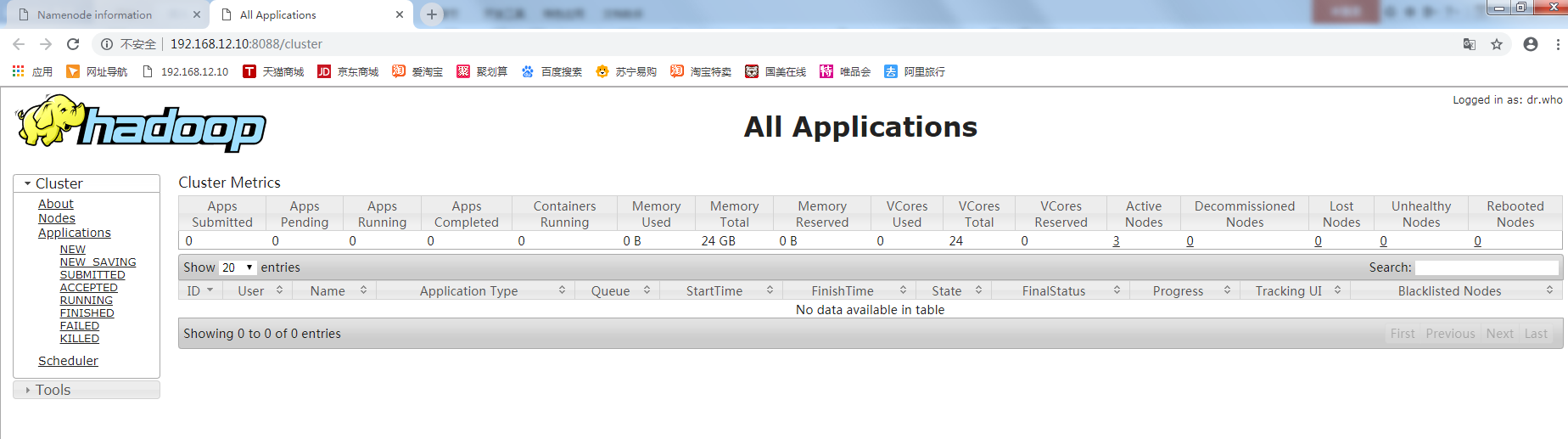
### 访问页面

http://192.168.12.10:50070 （HDFS管理界面）





http://192.168.12.10:8088 （MR管理界面）



### 配置ssh免登陆

#生成ssh免登陆密钥

#进入到我的home目录

cd ~/.ssh

ssh-keygen -t rsa （四个回车）

执行完这个命令后，会生成两个文件id\_rsa（私钥）、id\_rsa.pub（公钥）

将公钥拷贝到要免密登陆的目标机器上

ssh-copy-id localhost

---------------------------

ssh免登陆：

生成key:

ssh-keygen

(一直回车)（蔡）

复制从A复制到B上:

ssh-copy-id B

验证：

ssh localhost/exit，ps -e|grep ssh

ssh A #在B中执行

exit 退出远程连接（蔡）

操作步骤有：

|  |
| --- |
| #ssh localhost  #cd ~/.ssh  #ssh-keygen -t dsa -P '' -f ~/.ssh/id\_dsa  #cat ~/.ssh/id\_dsa.pub >> ~/.ssh/authorized\_keys  #ssh-copy-id node2  #ssh-copy-id node3  #ssh-copy-id node4 |

测试方法

|  |
| --- |
| #ssh node2  #eixt |

免密登录失败解决方法



systemctl restart sshd（把最后一行改下[centOS7下]）

## zookeeper分布式集群

1.选取三台服务器192.168.13.39，192.168.13.40，192.168.13.41。

2.解压zookeeper。

|  |
| --- |
| #mkdir -p /data/zookeeper/zookeeper\_cluster  #tar -zxvf zookeeper-3.4.6.tar.gz -C /data/zookeeper/zookeeper\_cluster |

3.在/data/zookeeper/zookeeper\_cluster/zookeeper-3.4.6目录下新建data，logs两个文件夹。

4.进入zookeeper-3.4.9/conf目录，把zoo\_sample.cfg文件复制一份名字改成zoo.cfg。

|  |
| --- |
| #cp zoo\_sample.cfg zoo.cfg |

5.修改zoo.cfg文件，需要修改以下几个地方：

|  |
| --- |
| dataDir=/data/zookeeper/zookeeper\_cluster/zookeeper-3.4.6/data  dataLogDir=/data/zookeeper/zookeeper\_cluster/zookeeper-3.4.6/logs  clientPort=2181  同时在文件末尾添加：  server.1=192.168.13.39:2888:3888  server.2=192.168.13.40:2888:3888  server.3=192.168.13.41:2888:3888 |

6.将配置好的zookeeper复制到另外两台服务器上。

|  |
| --- |
| #scp -r /data root@192.168.13.40:/  #scp -r /data root@192.168.13.41:/ |

7.分别在三台服务器的/data/zookeeper/zookeeper\_cluster/zookeeper-3.4.6/data目录下新建myid文件，内容分别为server.1，server.2，server.3后面的数字，例如：在192.168.13.39服务器上执行如下命令，另外两台服务器类似。

|  |
| --- |
| # echo "1" > myid  解释：新建文件myid，并且内容为1 |

8.分别进入三台服各器的/data/zookeeper/zookeeper\_cluster/zookeeper-3.4.6/bin目录下，启动服务。

|  |
| --- |
| # ./zkServer.sh start |

10.启动完成后，查看服务状态。

|  |
| --- |
| # ./zkServer.sh status |

结果为: 一主两从

## HBase完全分布式集群部署

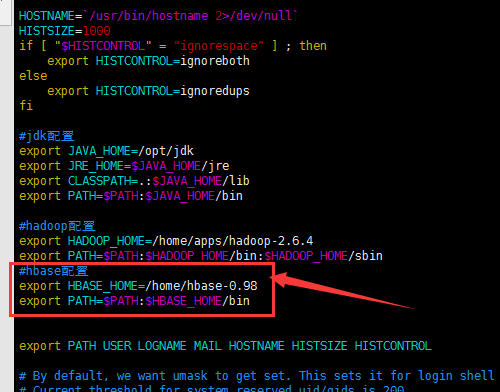
解压包

|  |
| --- |
| #tar zxvf hbase-0.98.12.1-hadoop2-bin.tar.gz -C /home/  #mv hbase-0.98.12.1-hadoop2 hbase-0.98 |

配置环境变量

|  |
| --- |
| #vim /etc/profile |

添加如下



|  |
| --- |
| export HBASE\_HOME=/home/hbase-0.98  export PATH=$PATH:$HBASE\_HOME/bin |

source /etc/profile（更新配置文件）

|  |
| --- |
| #vim /home/hbase-0.98/conf/hbase-env.sh |

修改如下

配置jdk和关闭HBase自带的Zookeeper,使用Zookeeper集群：

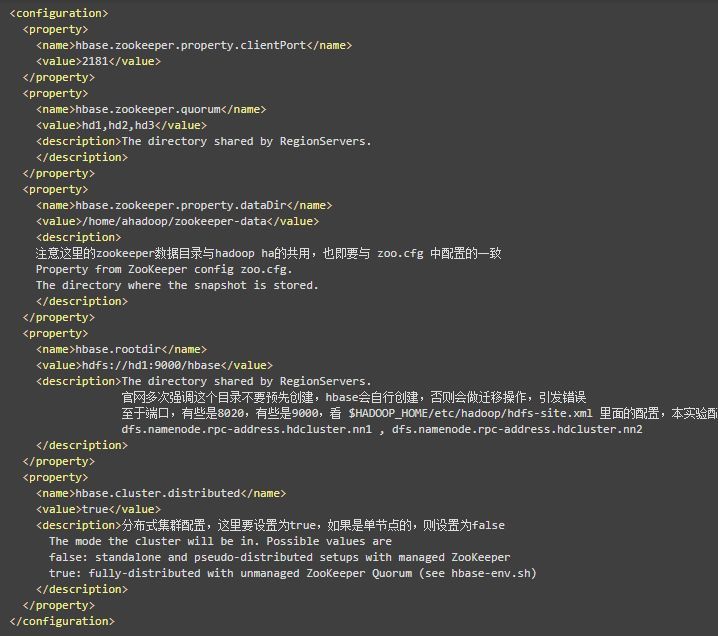
|  |
| --- |
| export JAVA\_HOME=/opt/jdk  export  HBASE\_MANAGES\_ZK=false |

修改hbase-site.xml

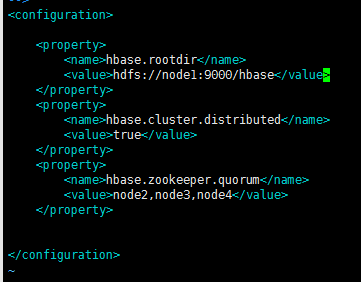
|  |
| --- |
| #vim /home/hbase-0.98/conf/hbase-site.xml |

|  |
| --- |
| <configuration>  　　<property>  　　　　<name>hbase.rootdir</name>  **<--在HDFS上创建一个干净的节点，用于存放元数据,教程上要求写**nameservices=bjsxt，但是我没有成功，所以还是写node1**-->**  　　　　<value>hdfs://node1:9000/hbase</value>  　　</property>  　　<property>  　　　　<name>hbase.cluster.distributed</name>  **<--是否开启分布式-->**  　　　　<value>true</value>  　　</property>  　　<property>  　　　　<name>hbase.zookeeper.quorum</name>  **<--对应的zookeeper集群，不用加端口-->**  　　　　<value>node2,node3,node4</value>  　　</property>  </configuration> |

别人这样写

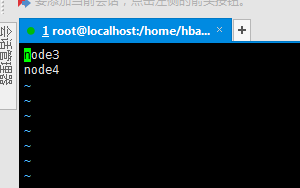


给为这样是可以正常运行了



编辑配置目录下面的文件regionservers. 命令：

|  |
| --- |
| # vi   regionservers |



|  |
| --- |
| node3 node4 |

backup-masters配置（注意backup-masters需要手动创建）内容为:

|  |
| --- |
| node02 |

把hadoop里面的 hdfs-site.xml 拷贝到

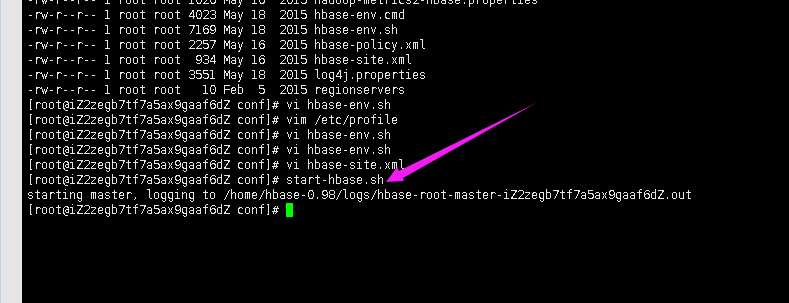
|  |
| --- |
| #cp /home/apps/hadoop-2.6.4/etc/hadoop/hdfs-site.xml /home/hbase-0.98/conf/ |

把Hbase复制到其他机器(复制3份)，命令如下：

|  |
| --- |
| #scp -r /home/apps/hadoop-2.6.4 [root@node2:/home/apps](mailto:root@node2:/home/apps/hadoop-2.6.4) |

在node01（Master节点） 执行 start-hbase.sh启动主节点，所以得都启动了。

|  |
| --- |
| #start-hbase.sh（启动）  #stop-hbase.sh（停止） |



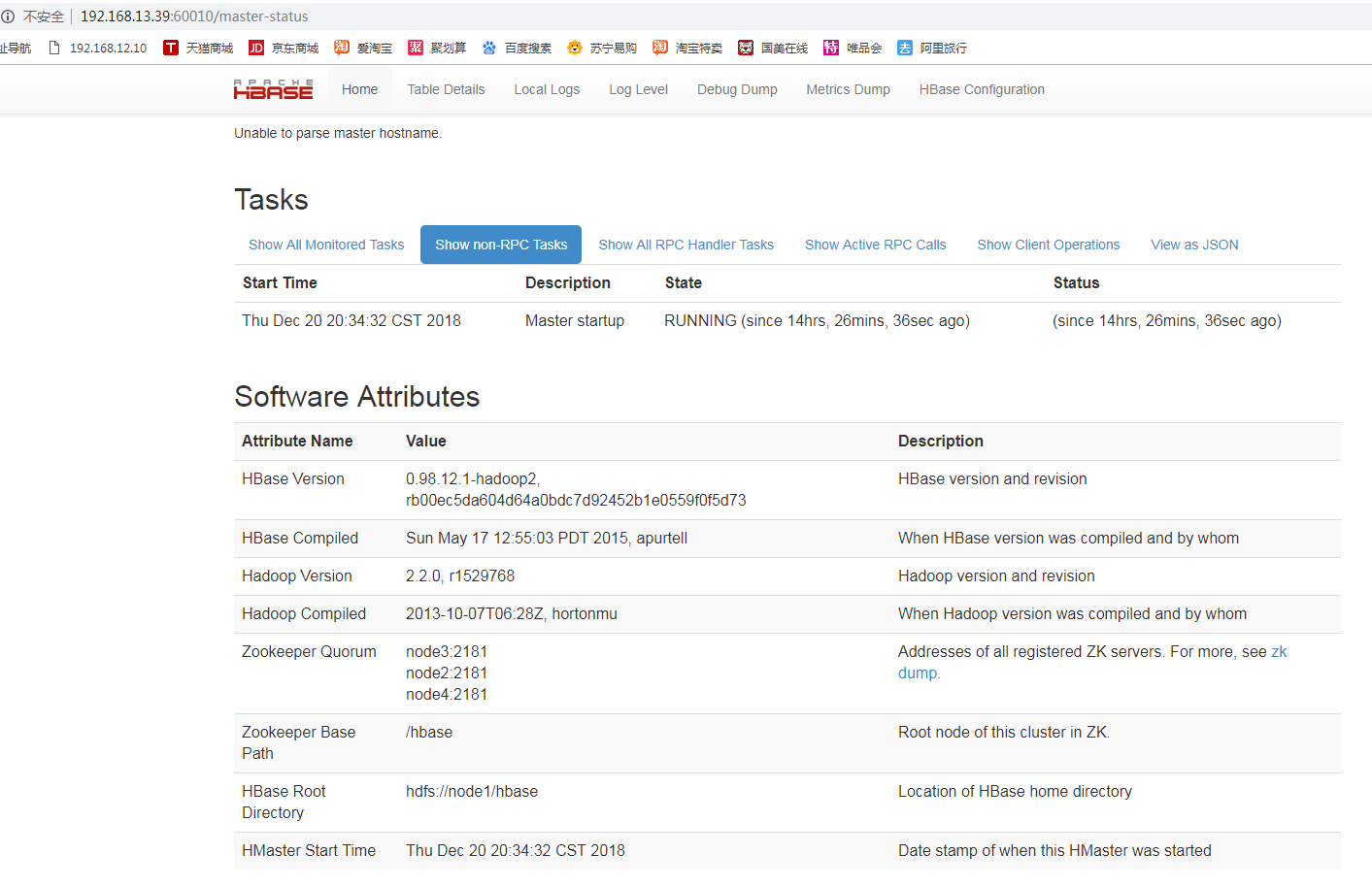
进入命令：

|  |
| --- |
| #hbase shell |

访问界面

http://192.168.12.10:60010（master-主）

http://192.168.13.39:60010（master-备）



备注:

建立软连接

ln -sf /home/hadoop-2.6.4/ hadoop

查看log日志

tail -n50 hadoop-root-namenode-localhost.localdomain.log

Hbase官方文档中写明，目前列族数量最优不超过3个