# 大数据hadoop3.1.3-zookeeper3.4.14-hbase2.2.4-CenOS 7伪分布式部署的高可用

<https://blog.csdn.net/wx2007xing/article/details/88710235>根据该文章修改补充

## 一 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



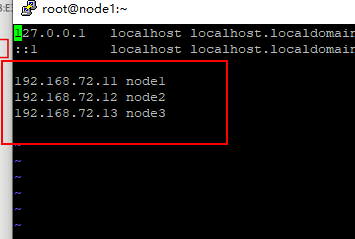
### 2、配置root账号的无密码登陆（可选）

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



### 5、防火墙关闭——生产环境不要这样，非常不安全

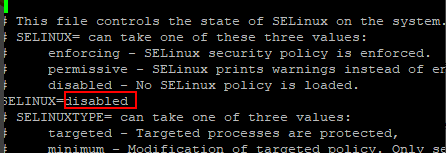
关： systemctl stop firewalld.service  
开机不启动： systemctl disable firewalld.service



### 6、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



### 7、创建组与用户

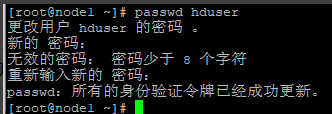
创建组，组名是 hadoop： groupadd hadoop

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

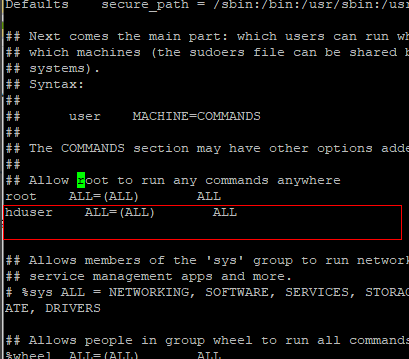
备注：当密码过于简单时，会有相关的安全提示，可以不理会，输入两次后，成功修改密码



### 8、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

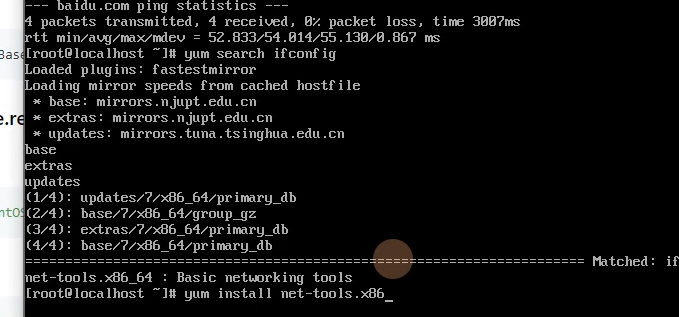
## 二 配置linux的工具

vi /etc/sysconfig/network-scripts/ifcfg-ens33

### 1、查看ip的工具

查看需要下载那个版本的工具

yum search ifconfig



显示需要安装

yum install net-tools.x86\_64

### 更改源 centOS

下载wget工具

yum install -y wget

# 进入yum源配置文件所在文件夹

cd /etc/yum.repos.d/

# 备份本地yum源

mv CentOS-Base.repo CentOS-Base.repo\_bak

# 获取国内yum源（阿里、163二选一）

wget -O CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-7.repo

# wget -O CentOS-Base.repo http://mirrors.163.com/.help/CentOS7-Base-163.repo

# 清理yum缓存

yum clean all

# 重建缓存

yum makecache

### 安装vim工具

yum install vim -y

### 时间同步

ln -sf /usr/share/zoneinfo/Asia/shanghai /etc/localtime

安装ntp

yum install -y ntp

将node1做为ntp服务修改ntp配置文件

vi /etc/ntp.conf 在后面写

server 127.127.1.0

fudge 127.127.1.0 stratum 10

重启ntp服务

/bin/systemctl restart ntpd.service

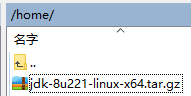
在其他系统上执行脚本同步时间

ntpdate node1

## 三 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

## 四 安装zookeeper集群

### 1、解压软件包到hduser目录下

使用hduser账号登陆

*# tar -zxvf zookeeper-3.4.14.tar.gz*

修改文件名为*zookeeper*

*mv zookeeper-3.1.2 hadoop*

### 配置环境变量

sodu vi /etc/profile

*export ZOOKEEPER\_HOME=/home/hduser/zookeeper*

*export PATH=$ZOOKEEPER\_HOME/bin:$PATH*

进入到zookeeper安装目录，并拷贝conf目录下zoo\_sample.cfg为****zoo.cfg****

# cp zoo\_sample.cfg zoo.cfg

****编辑zoo.cfg****

*#*

# 存放数据文件

dataDir=/home/hduser/zookeeper/dataDir

# 存放日志文件

dataLogDir=/home/hduser/zookeeper/dataLogDir

*# 添加如下内容*

server.1=node1:2888:3888

server.2=node2:2888:3888

server.3=node3:2888:3888

****在zookeeper目录下，创建data目录****

*# mkdir* dataDir 和 dataLogDir

****6、 在dataDir目录下创建myid文件，并分别添加1、2、3三个数字。示例如下****

*# 在h1下面创建myid*

*# vim myid*

*# 添加数字1*

1

*# 保存退出*

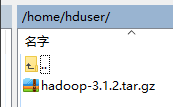
:wq

## 五 hadoop 3.1.3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



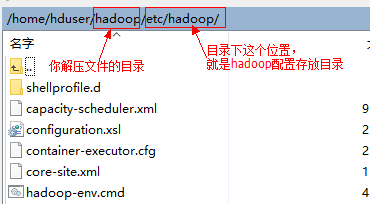
备注：当我们不知道解压后的文件名是什么时  
我们可以使用 ls  命令列出当前目录的文件名或文件夹名

### 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、进入配置存放目录



### 4、编辑core-site.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  <configuration>  <property>  <name>fs.defaultFS</name>  <value>hdfs://hadoop-cluster</value>  <description>集群名称</description>  </property>  <property>  <name>io.file.buffer.size</name>  <value>131072</value>  </property>  <property>  <name>hadoop.tmp.dir</name>  <value>/home/hduser/hadoop/tmp</value>  <desciription>A base for other temporsary directories.</desciription>  </property>  <property>  <name>ha.zookeeper.quorum</name>  <value>node1:2181,node2:2181,node3:2181</value>  <description>zookeeper集群</description>  </property>  </configuration> |

### 5、修改hdfs-site.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  <configuration>  <property>  <name>dfs.nameservices</name>  <value>hadoop-cluster</value>  <description>集群名称</description>  </property>  <property>  <name>dfs.ha.namenodes.hadoop-cluster</name>  <value>node11,node22</value>  <description>namenode名称，随便，可以是node11,node22，注意和下面保持一致</description>  </property>  <property>  <name>dfs.namenode.rpc-address.hadoop-cluster.node11</name>  <value>node1:9000</value>  <description>node11的RPC通信地址,注意务必是9000，不要用8020之类的，否则ui会有点问题</description>  </property>  <property>  <name>dfs.namenode.rpc-address.hadoop-cluster.node22</name>  <value>node2:9000</value>  <description>同上</description>  </property>  <property>  <name>dfs.namenode.http-address.hadoop-cluster.node11</name>  <value>node1:9870</value>  <description>ui地址及端口（原先是50070），随便</description>  </property>  <property>  <name>dfs.namenode.http-address.hadoop-cluster.node22</name>  <value>node2:9870</value>  <description>同上</description>  </property>  <property>  <name>dfs.namenode.shared.edits.dir</name>  <value>qjournal://node1:8485;node2:8485;node3:8485/hadoop-cluster</value>  <description>指定NameNode的edits元数据的共享存储位置。也就是JournalNode列表该url的配置格式：qjournal://host1:port1;host2:port2;host3:port3/journalId；journalId推荐使用nameservice，默认端口号是：8485</description>  </property>  <property>  <name>dfs.client.failover.proxy.provider.hadoop-cluster</name>  <value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>  <description>高可用失败自动切换代理服务</description>  </property>  <property>  <name>dfs.permissions.enabled</name>  <value>false</value>  <description>禁用权限</description>  </property>  <property>  <name>dfs.support.append</name>  <value>true</value>  <description>允许数据追加</description>  </property>  <property>  <name>dfs.replication</name>  <value>2</value>  </property>  <property>  <name>dfs.namenode.name.dir</name>  <value>/home/hduser/hadoop/dfs/namenode</value>  <description>namenode数据存放目录</description>  </property>  <property>  <name>dfs.datanode.data.dir</name>  <value>/home/hduser/hadoop/dfs/datanode</value>  <final>ture</final>  </property>  <property>  <name>dfs.journalnode.edits.dir</name>  <value>/home/hduser/hadoop/dfs/journalnode</value>  <description>journalnode数据存放目录</description>  </property>  <property>  <name>dfs.ha.automatic-failover.enabled</name>  <value>true</value>  <description>开启NameNode失败自动切换</description>  </property>  <property>  <name>dfs.webhdfs.enabled</name>  <value>ture</value>  <description>启用webhdfs</description>  </property>  <property>  <name>dfs.ha.fencing.ssh.connect-timeout</name>  <value>30000</value>  <description>配置sshfence隔离机制超时时间</description>  </property>  <property>  <name>dfs.ha.fencing.ssh.private-key-files</name>  <value>/root/.ssh/id\_rsa</value>  <description>使用sshfence隔离机制时需要ssh免登陆</description>  </property>  <property>  <name>dfs.ha.fencing.methods</name>  <value>sshfence  shell(/bin/true)  </value>  <description>配置隔离机制方法，多个机制用换行分割，即每个机制暂用一行</description>  </property>  </configuration> |

### 6、配置mapred-site.xml

|  |
| --- |
| <?xml version="1.0"?>  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  <configuration>  <property>  <name>mapreduce.framework.name</name>  <value>yarn</value>  </property>    </configuration> |

### 7、配置yarn-env.sh 加入java配置

export JAVA\_HOME=/usr/java/jdk

### 8、配置hadoop-env.sh 加入java配置

export JAVA\_HOME=/usr/java/jdk

### 9、配置yarn-site.xml

|  |
| --- |
| <?xml version="1.0"?>  <configuration>  <property>  <name>yarn.resourcemanager.ha.enabled</name>  <value>true</value>  <description>开启RM高可用</description>  </property>  <property>  <name>yarn.resourcemanager.cluster-id</name>  <value>yrc</value>  <description>指定RM的cluster id</description>  </property>  <property>  <name>yarn.resourcemanager.ha.rm-ids</name>  <value>rm1,rm2</value>  <description>指定RM的名字</description>  </property>  <property>  <name>yarn.resourcemanager.hostname.rm1</name>  <value>node1</value>  <description>指定RM的地址</description>  </property>    <property>  <name>yarn.resourcemanager.hostname.rm2</name>  <value>node2</value>  <description>指定RM的地址</description>  </property>  <property>  <name>yarn.resourcemanager.zk-address</name>  <value>node1:2181,node2:2181,node3:2181</value>  <description>zookeeper集群</description>  </property>  <property>  <name>yarn.nodemanager.aux-services</name>  <value>mapreduce\_shuffle</value>  <description></description>  </property>  <property>  <name>yarn.log-aggregation-enable</name>  <value>true</value>  <description></description>  </property>  <property>  <name>yarn.log-aggregation.retain-seconds</name>  <value>86400</value>  <description></description>  </property>  <property>  <name>yarn.resourcemanager.recovery.enabled</name>  <value>true</value>  <description>启用自动恢复</description>  </property>  <property>  <name>yarn.resourcemanager.store.class</name>  <value>org.apache.hadoop.yarn.server.resourcemanager.recovery.ZKRMStateStore</value>  <description>制定resourcemanager的状态信息存储在zookeeper集群上</description>  </property>  </configuration> |

### 10、编辑workers文件，以前是slaves文件

node1

node2

node3

### 11、编辑sbin目录下的start-dfs.sh 和stop-dfs.sh并添加如下内容

# vim start-dfs.sh

HDFS\_JOURNALNODE\_USER=root

HDFS\_DATANODE\_USER=hduser

HDFS\_DATANODE\_SECURE\_USER=hdfs

HDFS\_ZKFC\_USER=hduser

HDFS\_NAMENODE\_USER=hduser

HDFS\_SECONDARYNAMENODE\_USER=hduser

### 12、编辑sbin目录下的start-yarn.sh和stop-yarn.sh并添加如下内容

# vim start-yarn.sh

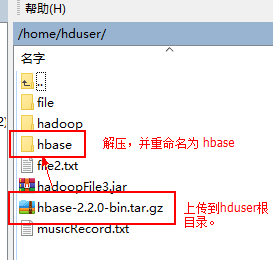
YARN\_RESOURCEMANAGER\_USER=hduser

HADOOP\_SECURE\_DN\_USER=yarn

YARN\_NODEMANAGER\_USER=hduser

## 六 配置hbase

### 1、准备阶段（node1,hduser账号）

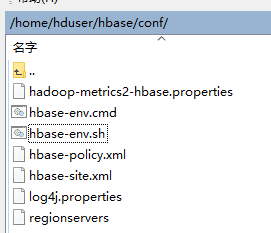


解压命令 tar -zxvf hbase-2.2.4-bin.tar.gz

重命名 mv hbase-2.2.4 hbase

### 2、配置（node1,hduser账号）

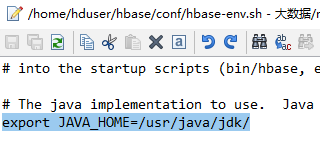
进入 /home/hduser/hbase/conf 目录



#### 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://hadoop-cluster/hbase</value>  </property>  <property>  <name>hbase.cluster.distributed</name>  <value>true</value>  <property>  <name>hbase.zookeeper.property.clientPort</name>  <value>2181</value>  </property>  </property>  <property>  <name>hbase.zookeeper.quorum</name>  <value>node1,node2,node3</value>  </property>  <property>  <name>hbase.zookeeper.property.dataDir</name>  <value>/home/hduser/zookeeper/data</value>  </property>  <property>  <name>hbase.unsafe.stream.capability.enforce</name>  <value>false</value>  </property>  </configuration> |

#### 2.3、编辑hbase-env.sh

# vim hbase-env.sh

# 主要是配置JAVA\_HOME，同hadoop-env.sh

export JAVA\_HOME=/usr/java/jdk/

# 可选

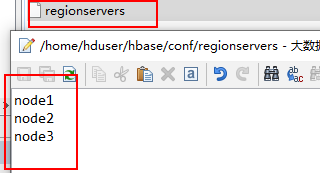
export HBASE\_CLASSPATH=$HBASE\_CLASSPATH:/usr/local/software/hadoop-3.1.2/etc/hadoop

# 关闭自身zookeeper，采用外部的zookeeper

export HBASE\_MANAGES\_ZK=false

#### 2.4、配置regionservers

在文件里，加入 node1 node2 node3

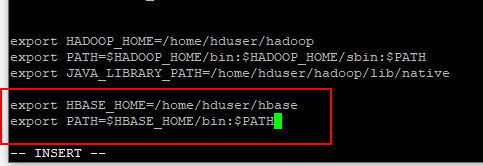


#### 2.5、设置环境变量（所有的主机node1,node2,node3都需要配置）

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



#### 2.6、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

#### 2.7、执行一下命令

让之前的环境变量配置生效

source /etc/profile

#### 2.8、将hdfs-site.xml映射到conf下hdfs-site.xml

# ln -s /usr/local/software/hadoop-3.1.2/etc/hadoop/hdfs-site.xml ./hdfs-site.xml

#### **2.9、配置Hmaster高可用**

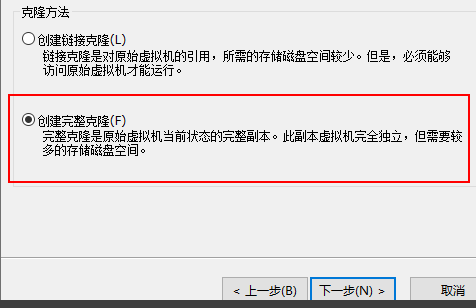
# vim backup-masters

h2

*至此Hbase配置完毕*

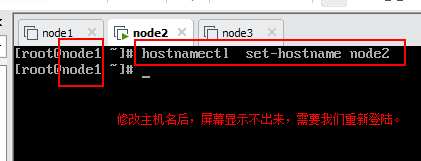
## 七 关要克隆2台(先把node1关机)

采用完整克隆方式，把node1克隆两次。分别为node2、node3

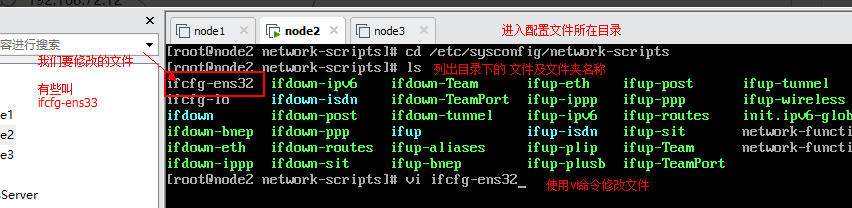


### 4.1、针对克隆机node2

**修改主机名**



如图，修改文件

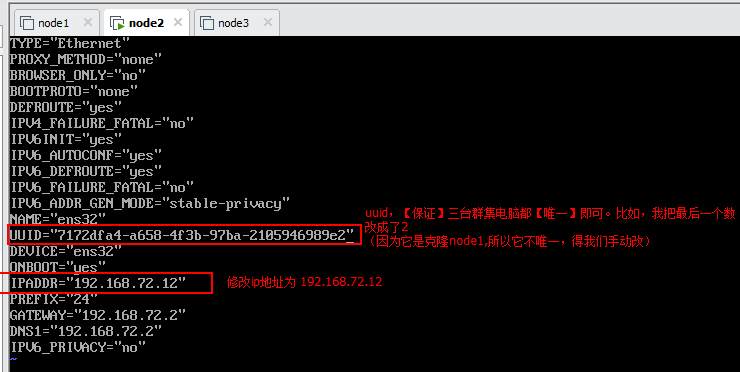


cd /etc/sysconfig/network-scripts

vi ifcfg-ens33

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



备注：将BOOTPROTO=static

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、永久修改主机名

输入 vi /etc/sysconfig/network 内容如下：

NETWORKING=yes

HOSTNAME=node1

#HOSTNAME每台服务设置成自己的主机名

然后重启：reboot

## 九 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

没有配置好在克隆将文件复制一下（zookeeper,hadoop,hbase）

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/

## 十 启动

由于已经配置了环境变量，所以任意位置执行以下命令都可以

关闭防火墙

systemctl stop firewalld

查看防火墙的状态

systemctl status firewalld

停止firewall

systemctl stop firewalld.service

禁止firewall 开机启动

systemctl disable firewalld.service

### 1、 启动zookeeper（h1 h2 h3）节点

# 启动

# zkServer.sh start

# 查看状态

# zkServer.sh status

ZooKeeper JMX enabled by default

Using config: /usr/local/software/zookeeper-3.4.13/bin/../conf/zoo.cfg

Mode: follower

### 2、启动journalnode（h1 h2 h3）节点

# hdfs --daemon start journalnode

# 或者(不推荐)

# hadoop-daemon.sh start journalnode

### 3、 格式化zkfc和namenode（h1）节点

# 格式化zkfc

# hdfs zkfc -formatZK

# 格式化namenode（如果在hosts没有配置 0.0.0.0 主机名在这一步的时候可能会报错）

# 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

### 4、 启动hadoop

# 下面这个命令会同时启动所有服务（当然您也可以单独启动服务）

# start-all.sh

### 5、 启动另一个namenode

# 同步fsimage

# hdfs namenode -bootstrapStandby

# 启动namenode

# hdfs --daemon.sh start namenode

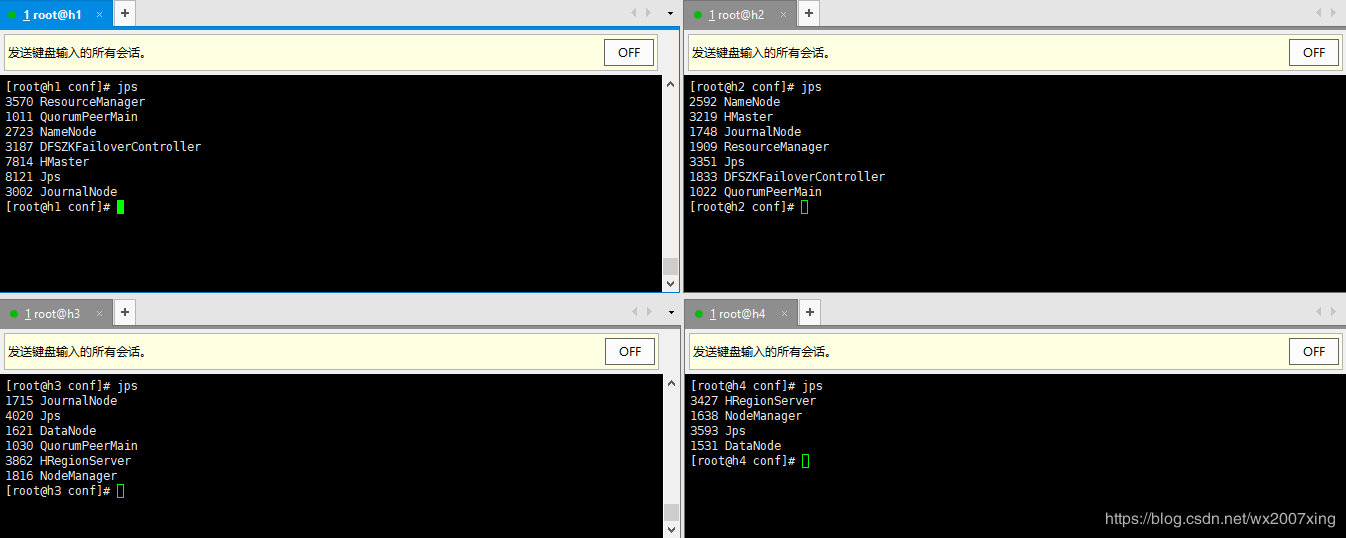
### 6、 启动hbase

# 注意，要在hadoop已经完全启动后再启动

# start-hbase.sh

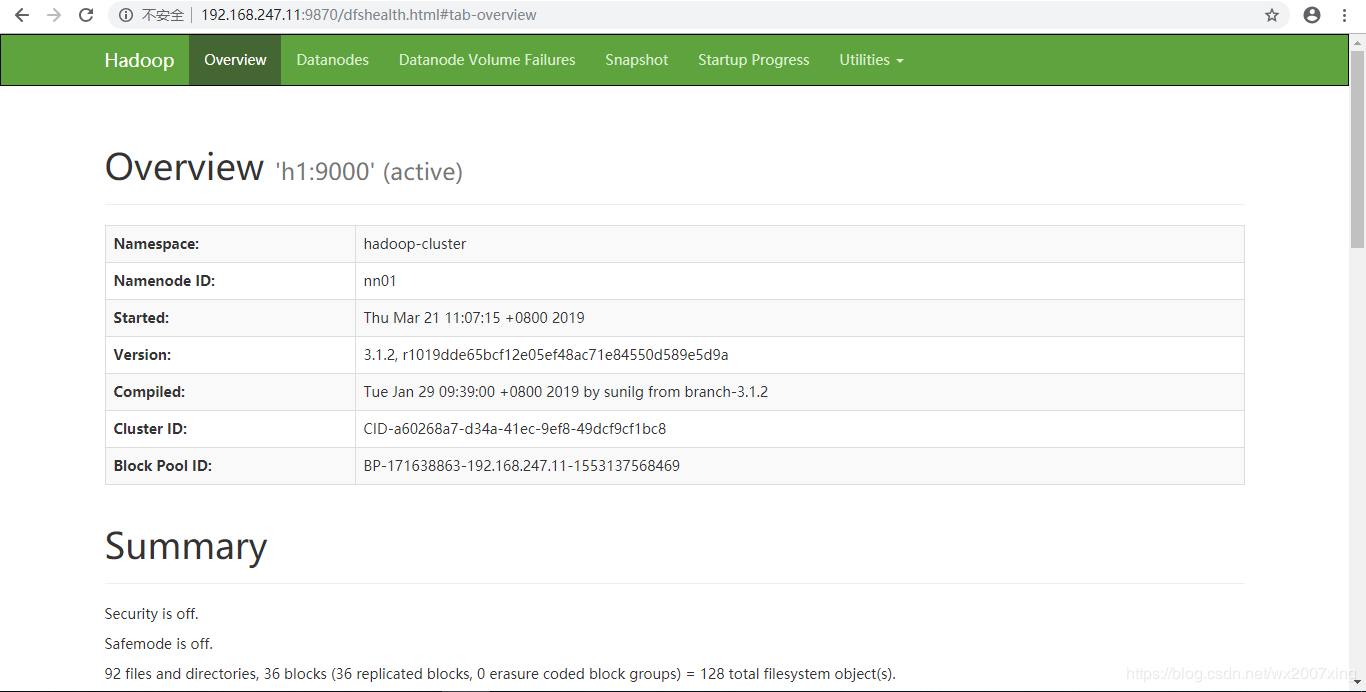
## 十一 最终效果

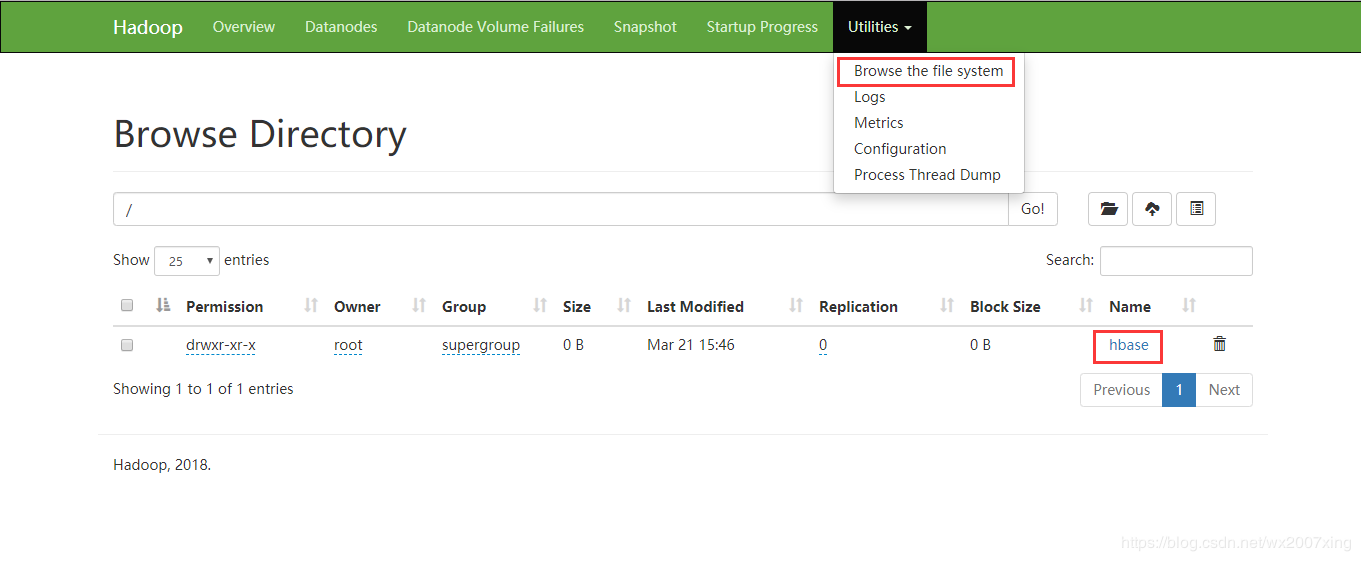
****1、 jps查看所有进程效果****



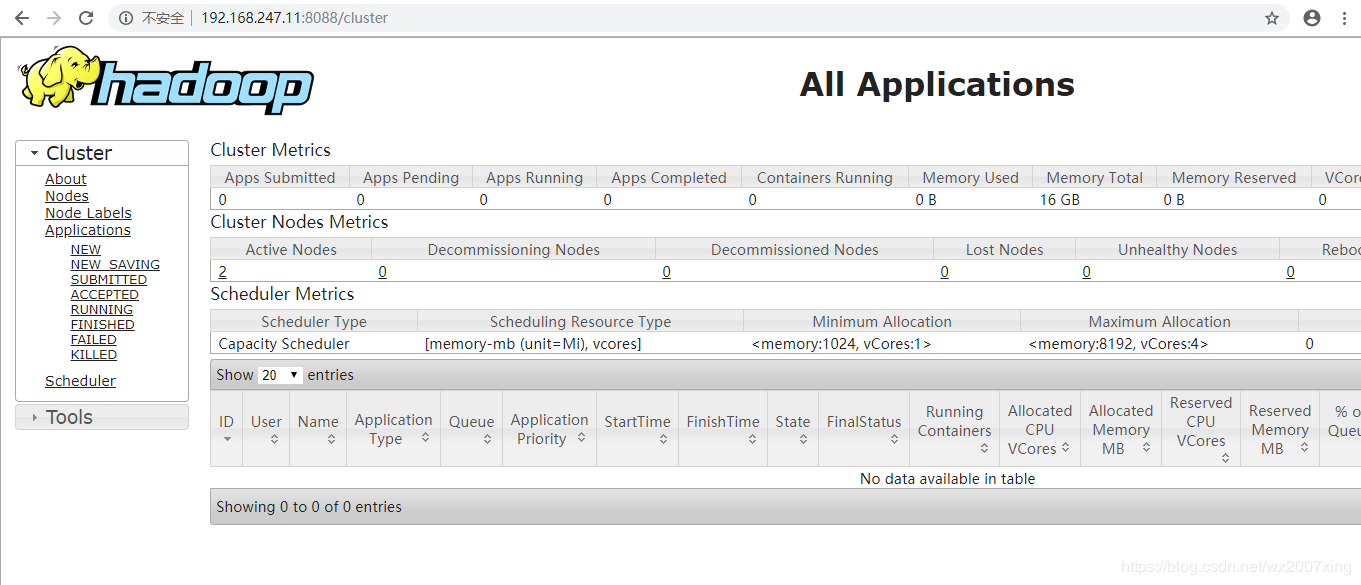
****2、 UI效果****

hadoop地址：IP:9870

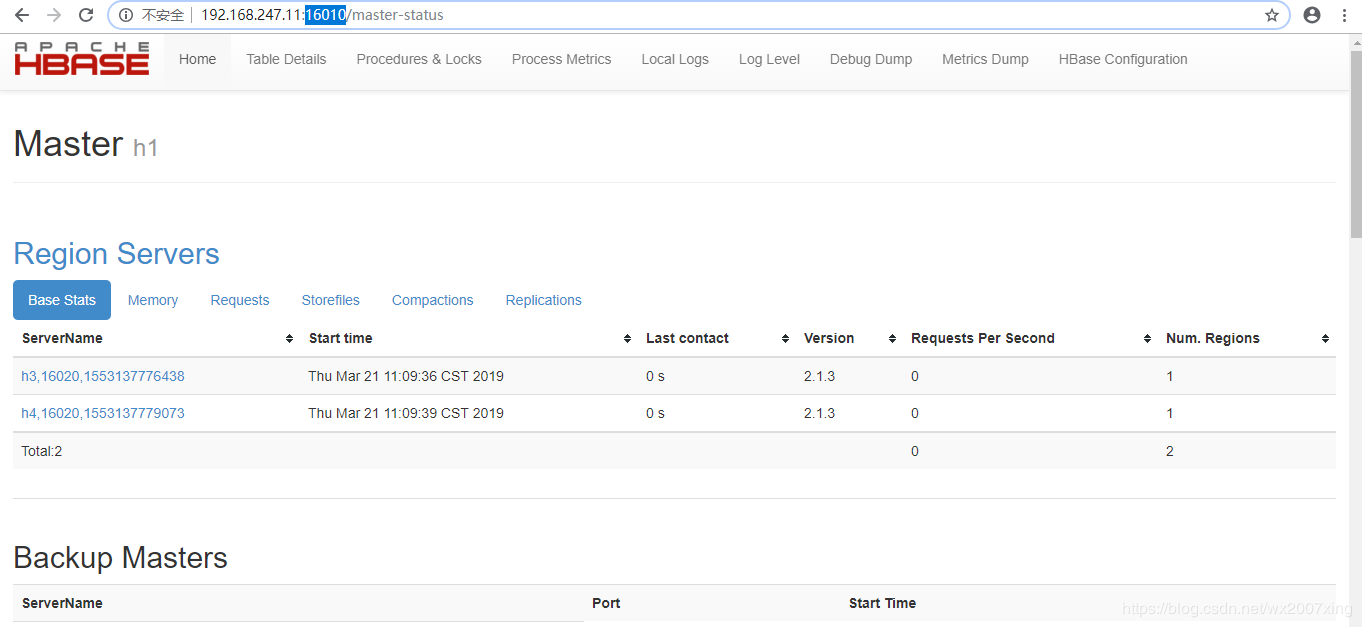




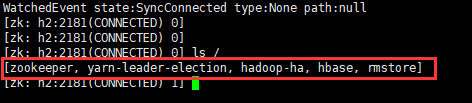
Yarn地址：IP:8088



HBase地址：IP:16010



ZK元数据



完