注意：java，Hadoop，Spark之间存在版本会兼容问题

**部分截图中的java是jdk-17\_linux-x64\_bin.tar，但实际上与spark-3.2.1-bin-hadoop3.2.gz和hadoop-3.3.1.tar.gz都不兼容，可用jdk-8u321-linux-x64.tar.gz。**

使用的包为（若截图的包名和这里不一致，那是因为我后来更新软件包后不方便重新截图，总之，安装的包以这里为准）：

java：jdk-8u321-linux-x64.tar.gz

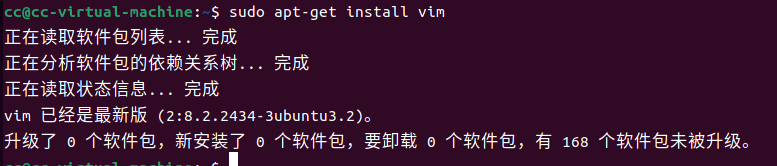
Hadoop：hadoop-3.3.1.tar.gz

zookeeper：apache-zookeeper-3.8.0-bin.tar.gz

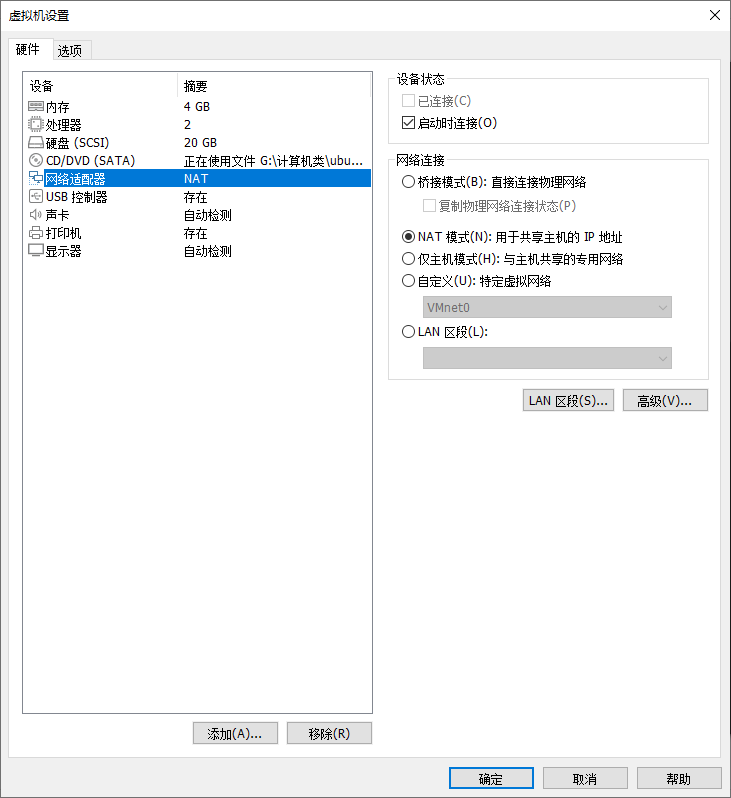
Spark：spark-3.2.1-bin-hadoop3.2.gz

使用ubuntu21.10时，刚开始先联网用sudo apt-get update更新一下软件包

安装vim命令：sudo apt-get install vim



网络是NAT模式

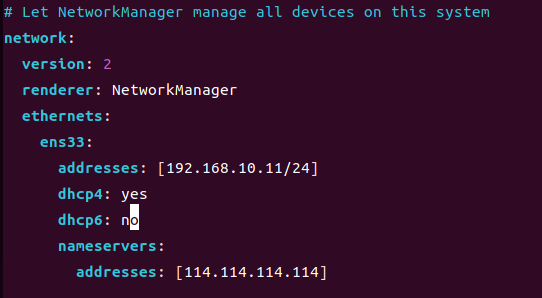


配置网络命令：

sudo vim /etc/netplan/…….yaml



进入之后编辑成：



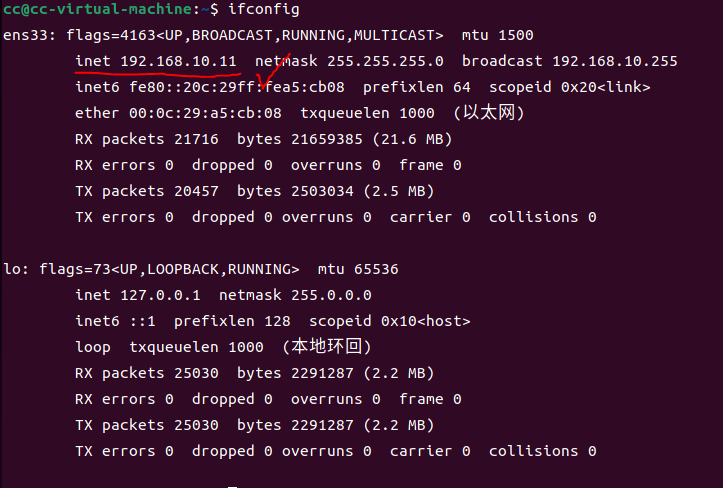
然后“:wq”保存退出

重启网卡：

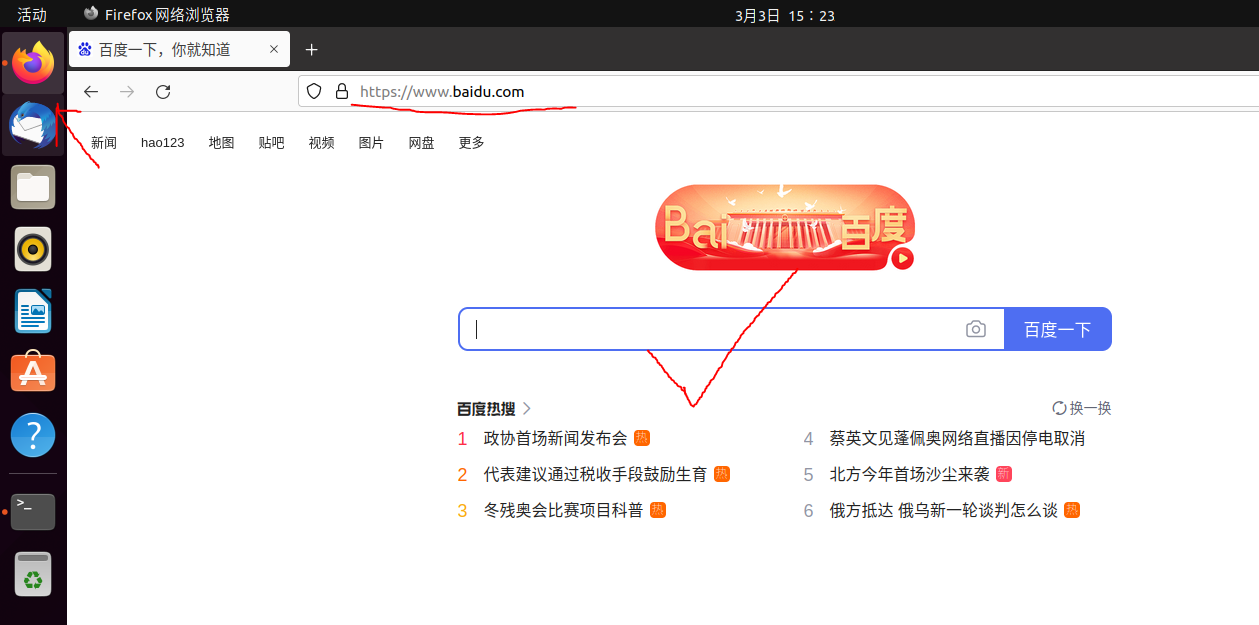
sudo netplan apply



可以用ifconfig查看网络是不是静态ip192.168.10.11



打开浏览器上百度看看是否能正常上网

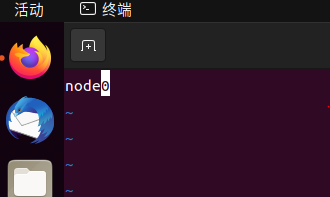


修改主机名：

sudo vim /etc/hostname



进去后编辑为：node0



编辑好后用“:wq”保存退出

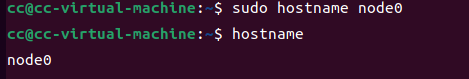
这个主机名应该是重启才生效的

如果想让主机名不重启就改为node0的话

可以用sudo hostname node0命令修改

hostname命令可以查看当前主机名

看到当前终端还是cc@cc-virtual-machine:不用管，重启之后才会变成cc@node0:

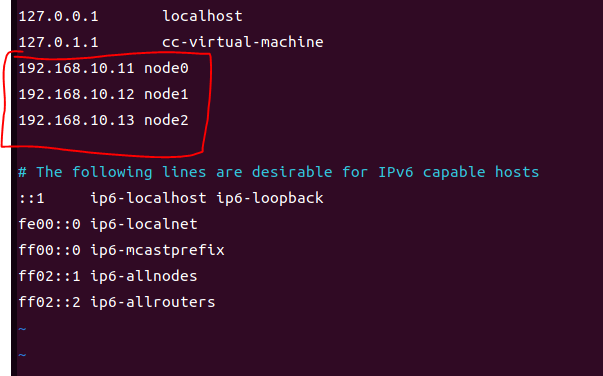


配置主机名与IP地址的映射信息：

sudo vim /etc/hosts



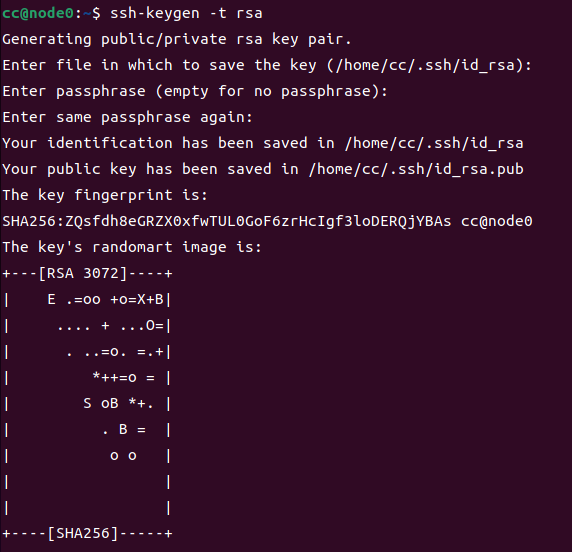
进去后编辑，增加这3条映射关系



然后“:wq”保存退出

用ssh-keygen -t rsa生成密钥，过程一直按enter键即可

（这里我因为重启过电脑，所以cc@cc-virtual-machine变成了cc@node0，不必理会）



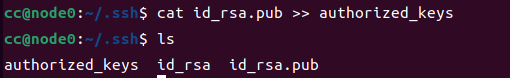
cd ~/.ssh进入密钥生成后的存放目录“~./ssh”





将公钥id\_rsa.pub复制到当前目录的/authorized\_keys列表中（如果当前目录没有authorized\_keys文件，则会自动创建）：

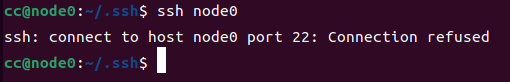
cat id\_rsa.pub >> authorized\_keys



ll查看当前目录下的文件与对应权限等，用命令“chmod 600 authorized\_keys”修改文件authorized\_keys的权限



使用命令ssh node0验证是否能够远程登录node0，如果出现类似下图的“Connection refused”拒绝连接，可能是还没安装openssh-server

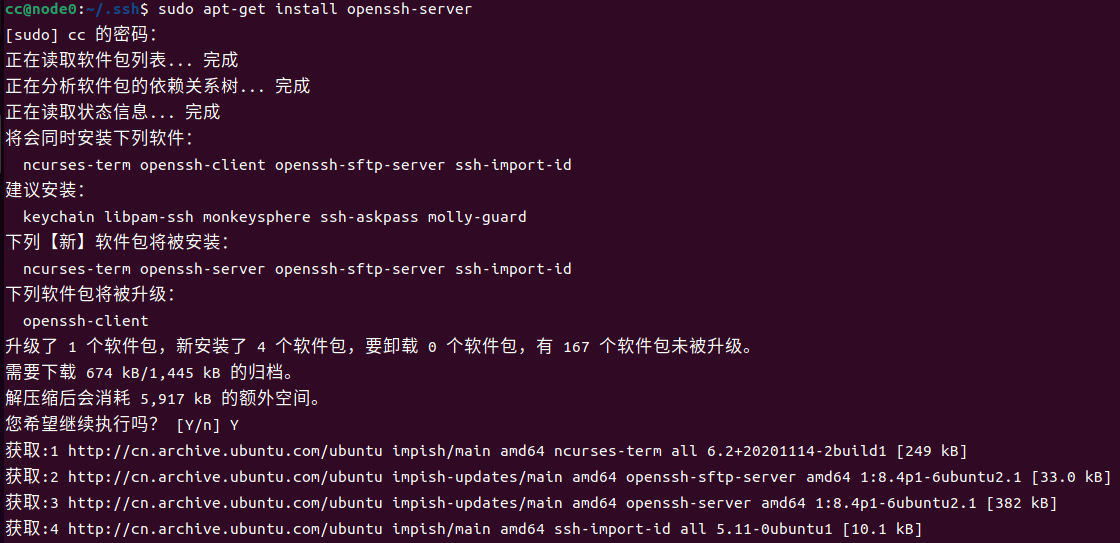


用命令ps -e |grep ssh检查是否安装了openssh-server



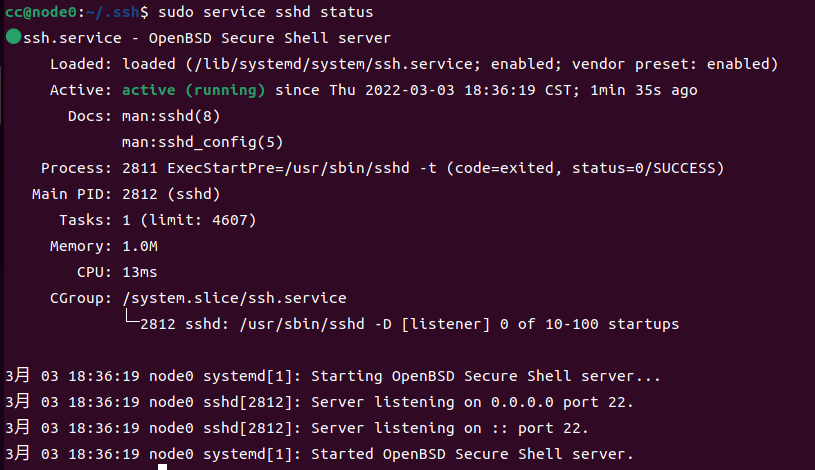
安装ssh服务：

sudo apt-get install openssh-server



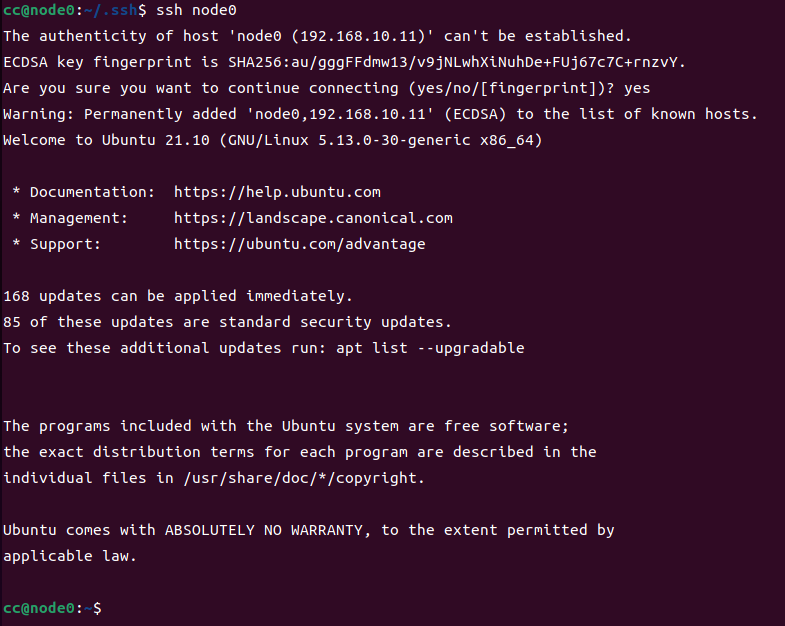
查看ssh服务启动状态：

sudo service sshd status



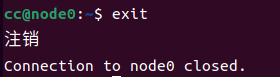
绿色的active（running）表示运行中

再试一次ssh node0：

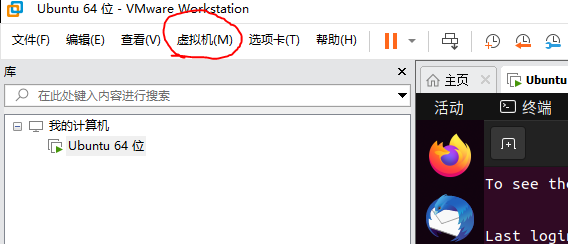


退出当前远程登录：

exit



点击VMware菜单栏中的“虚拟机”，然后点“设置”



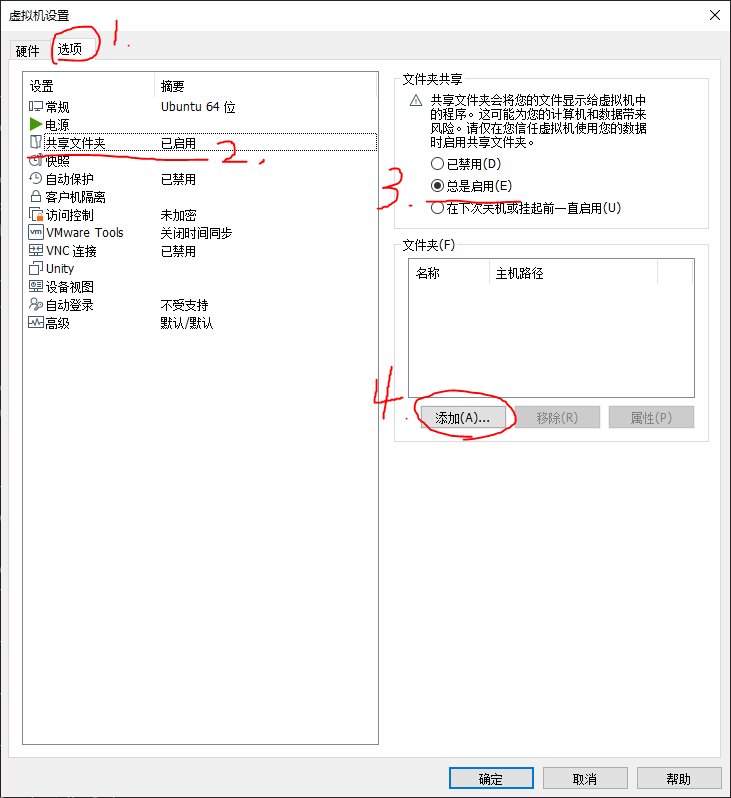
弹出来的框（虚拟机设置），

第一步，点“选项”

第二步，点共享文件夹

第三步，点总是启用

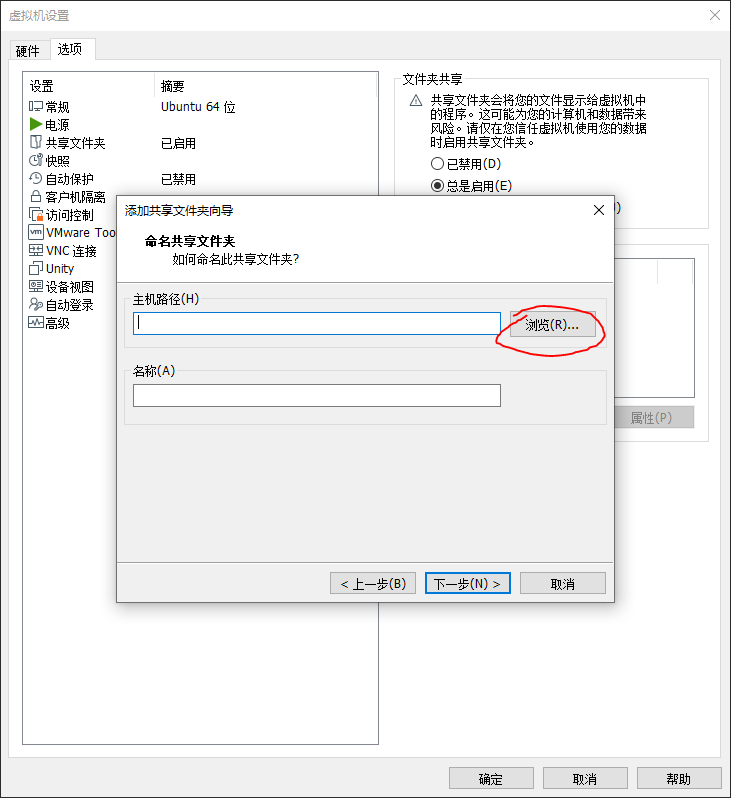
第四步，点添加



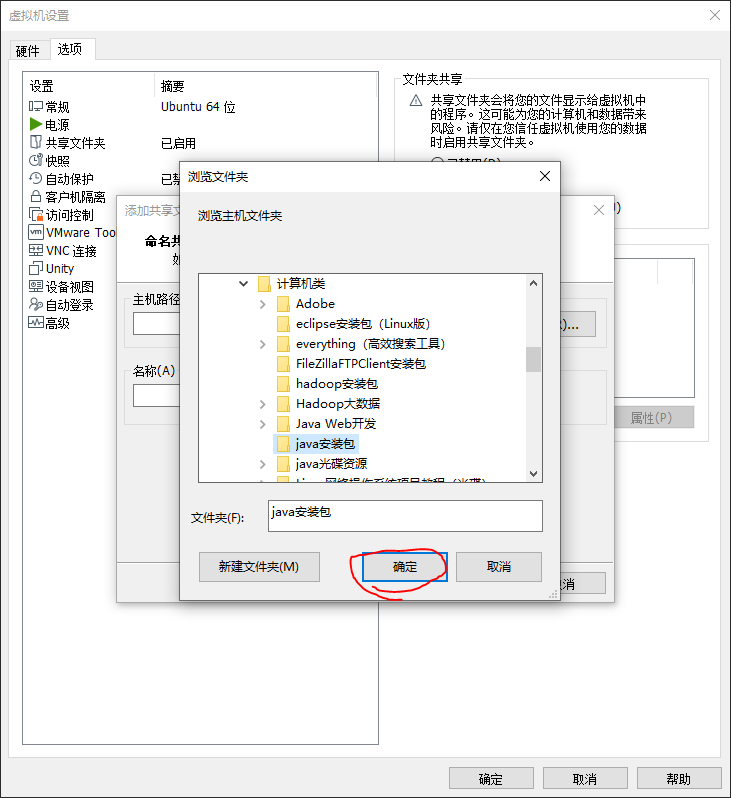
弹出的框，点下一步



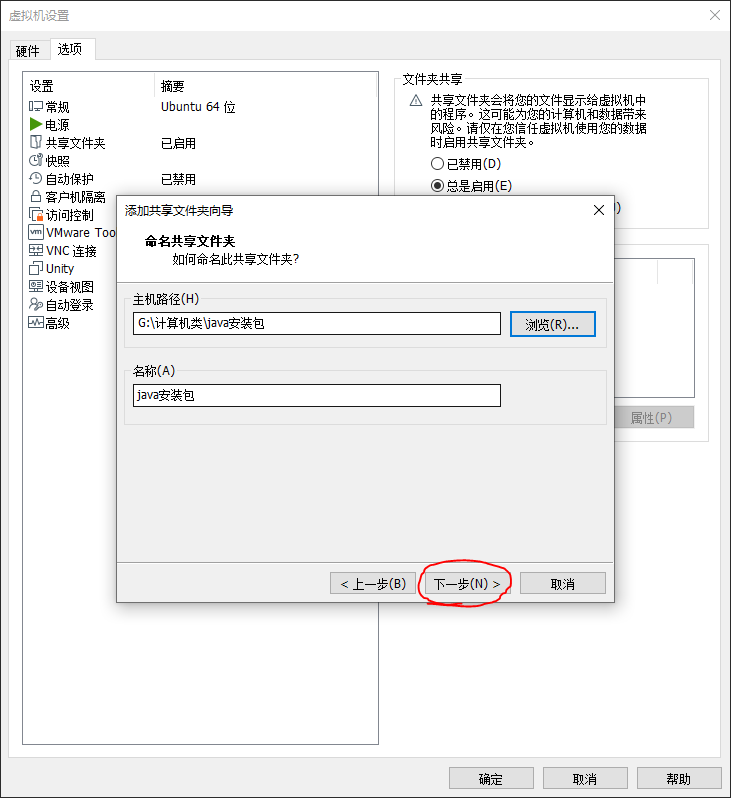
然后点浏览



找到java的安装包的储存位置后点确认



点下一步



点完成



点确认



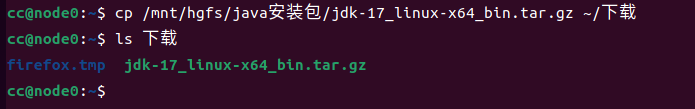
在ubuntu中查看共享文件夹里的jdk安装包：

ls /mnt/hgfs/java安装包



拷贝jdk安装包到当前用户目录的“下载”文件夹中：

cp /mnt/hgfs/java安装包/jdk-17\_linux-x64\_bin.tar.gz ~/下载



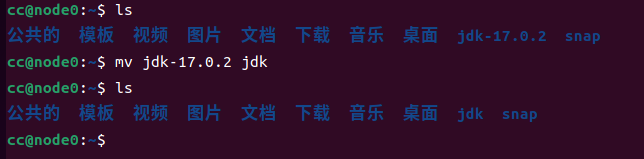
解压jdk安装包到当前目录（~）下，-C 路径（解压到指定目录）：

tar -zxvf ./下载/jdk-17\_linux-x64\_bin.tar.gz -C ./



解压完成后把解压出来的文件夹jdk-17.0.2改名为jdk:

mv jdk-17.0.2 jdk



修改环境配置文件：

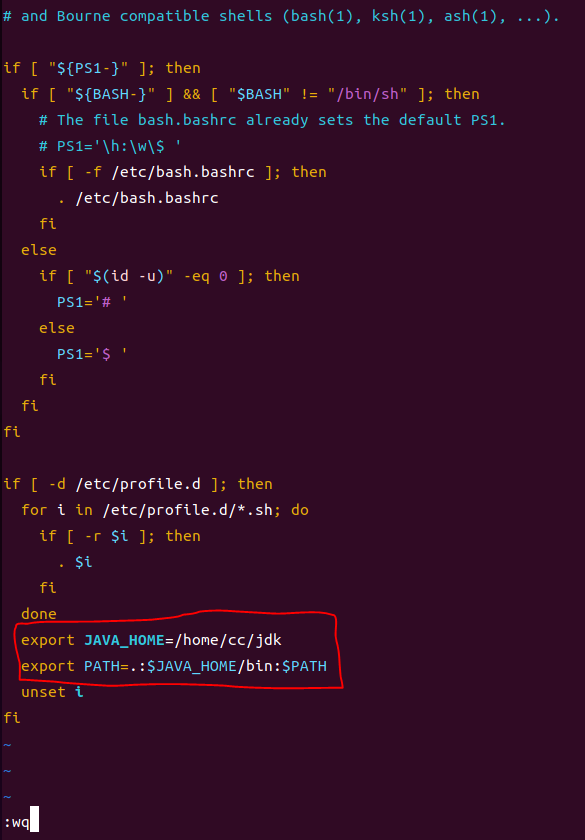
sudo vim /etc/profile



在如图所示位置加入这两行：

export JAVA\_HOME=/home/cc/jdk

export PATH=.:$JAVA\_HOME/bin:$PATH

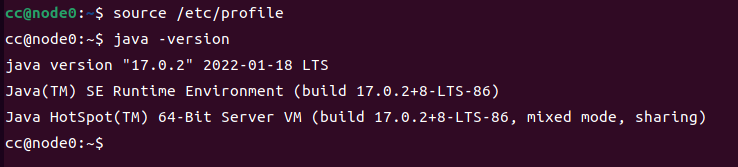


然后“:wq”保存退出

用source命令使Shell读入指定的Shell程序文件并依次执行文件中的所有语句：

source /etc/profile

然后用java -version命令查看java版本同时判断java是否安装成功，如下图所示即成功

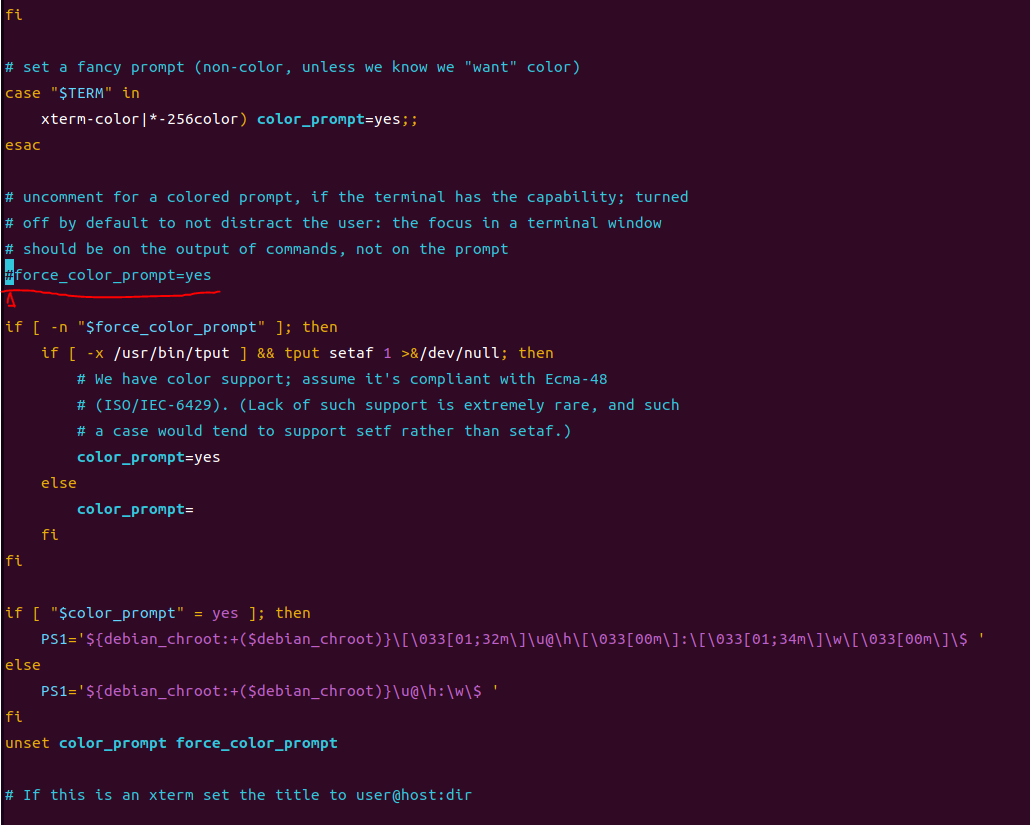


如果“cc@node0:”变成白色，修改配置文件.bashrc：

vim .bashrc



找到“#force\_color\_prompt=yes”把前面的“#”号去掉后保存退出

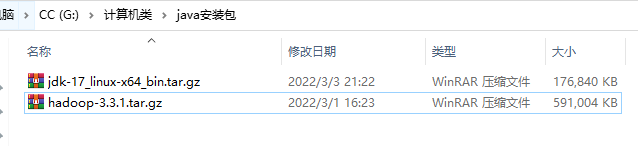


用source命令重新执行.bashrc配置文件：

source ./.bashrc



因为前面把java安装包的位置设置为共享文件夹，所以把hadoop安装包放到java安装包这里，当然，也可以选择重新设置共享文件夹路径到hadoop安装包的位置



可以看到hadoop可以在ubuntu中找到了

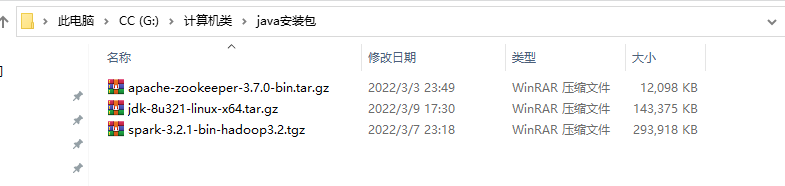


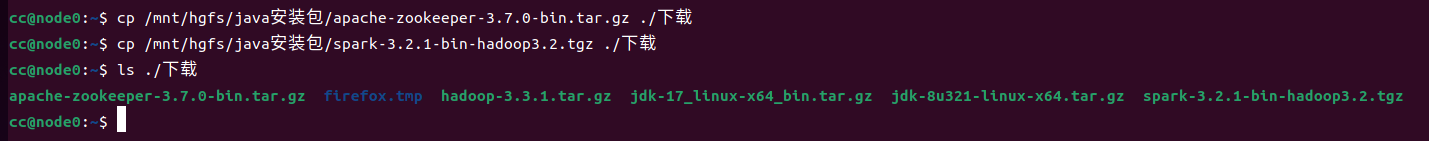
拷贝hadoop安装包到当前用户目录的“下载”文件夹里：

cp /mnt/hgfs/java安装包/Hadoop-3.3.1.tar.gz 下载

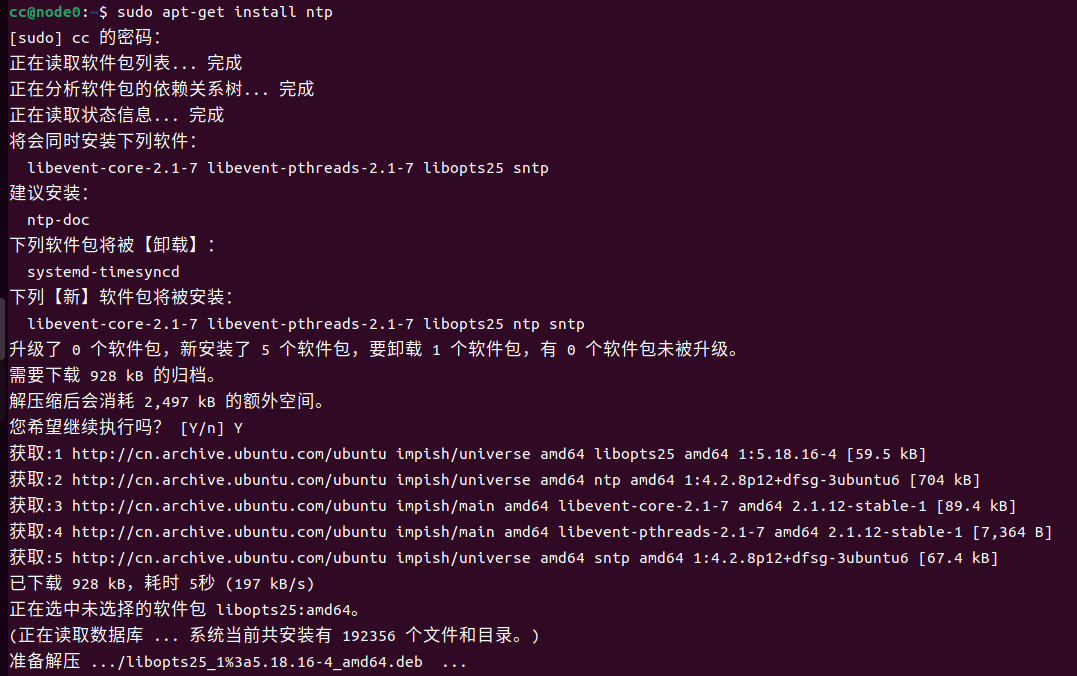


ZooKeeper安装包和Spark安装包同理：



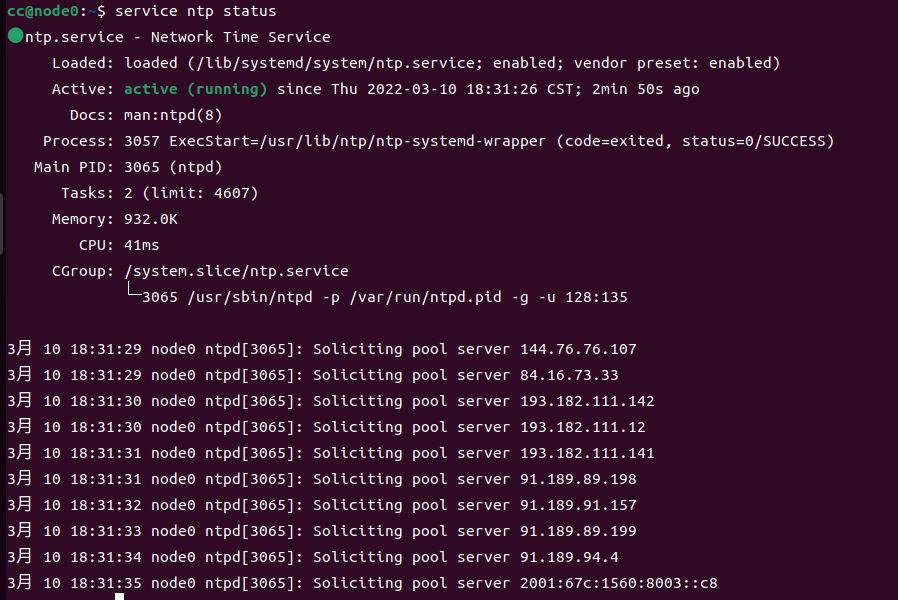


安装ntp时间同步（需要联网）：



查看ntp服务状态：

service ntp status

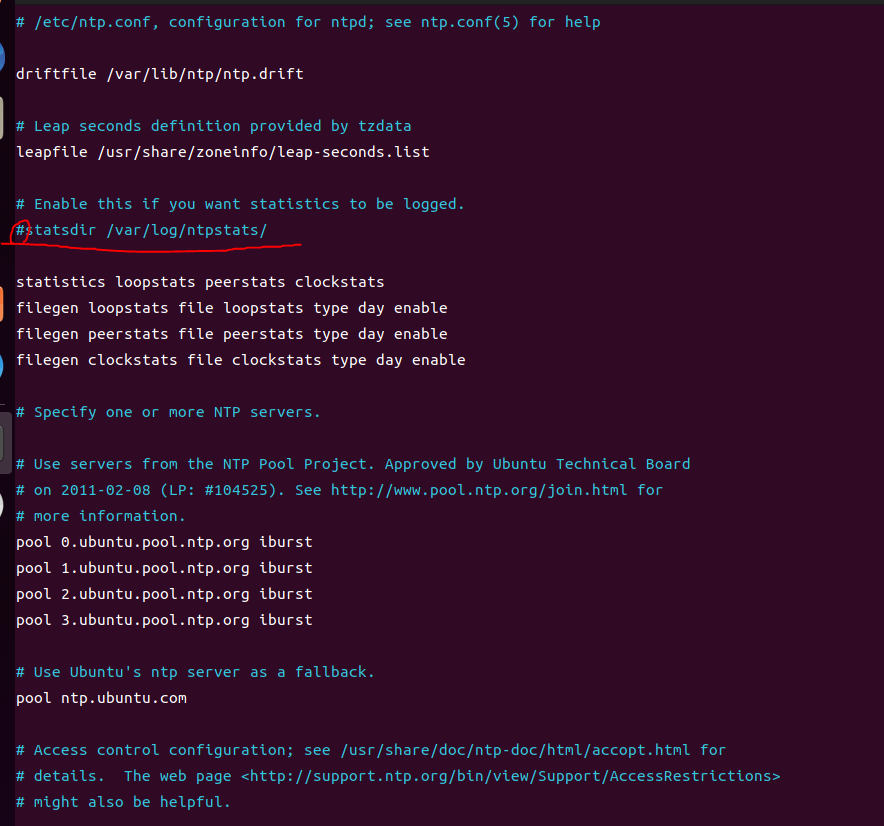


配置ntp：

sudo vim /etc/ntp.conf



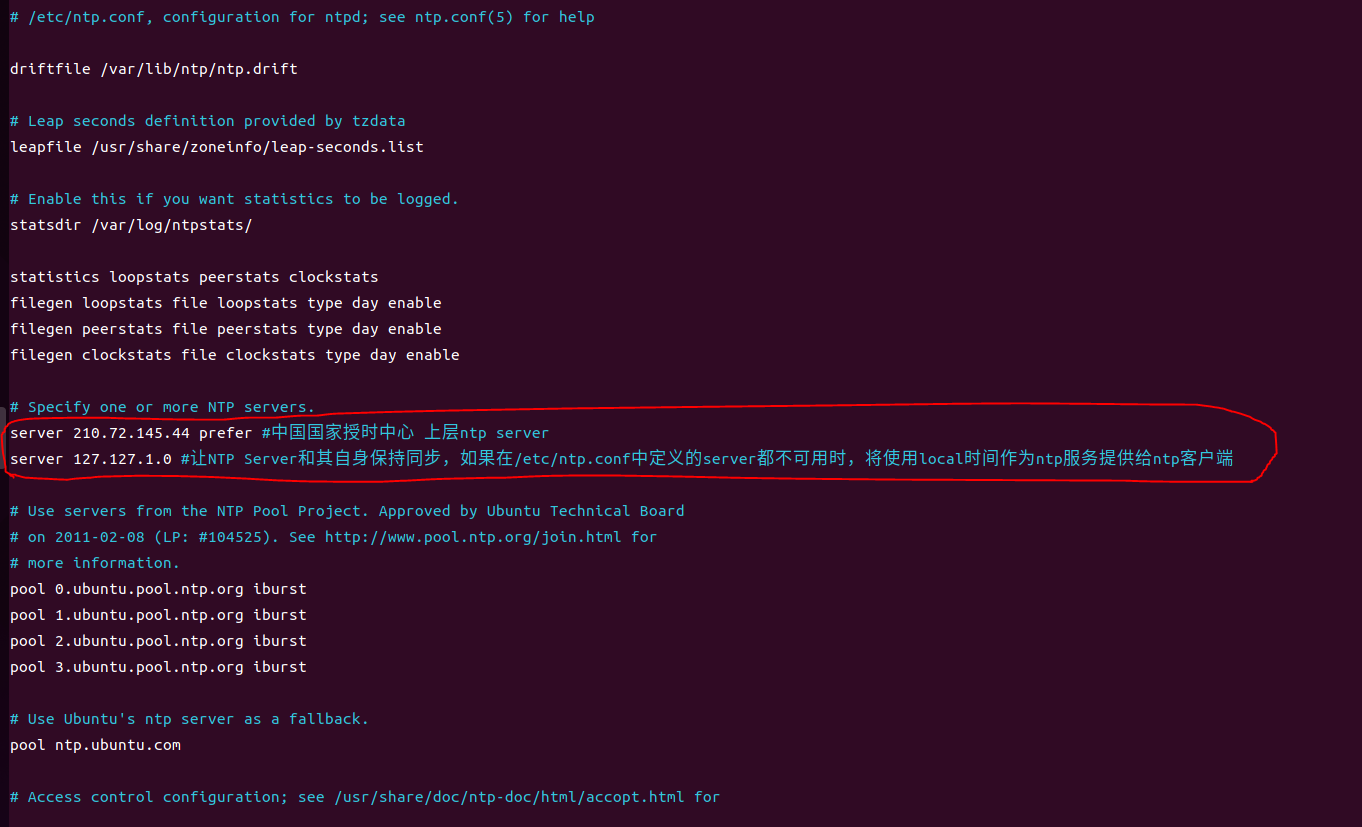
去掉“statsdir /var/log/ntpstats”前面的“#”号后，即可在/var/log/ntpstats/中查看ntp的日志信息



并添加以下配置：

server 210.72.145.prefer

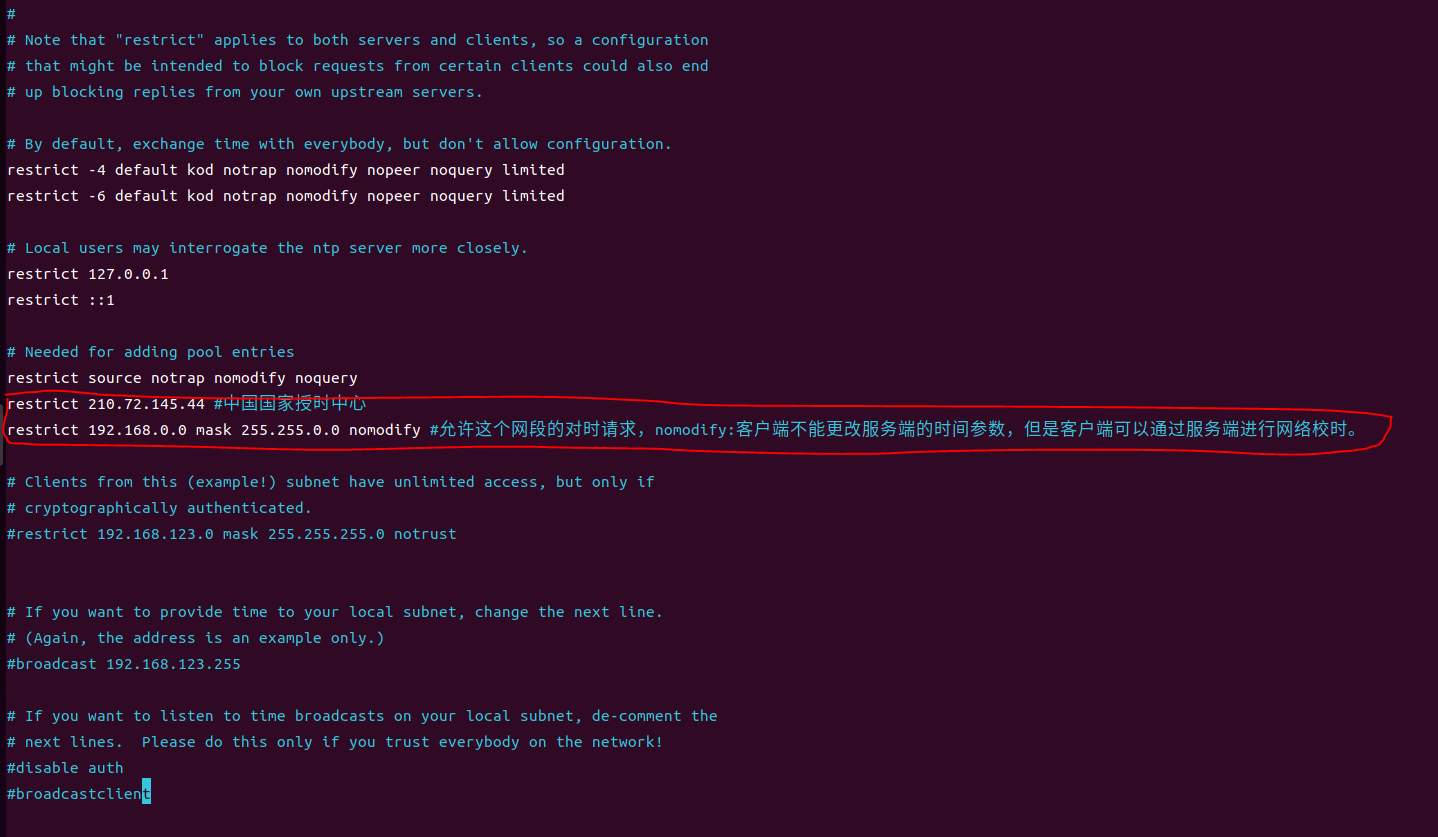
server 127.127.1.0



再添加以下配置：

restrict 210.72.145.44

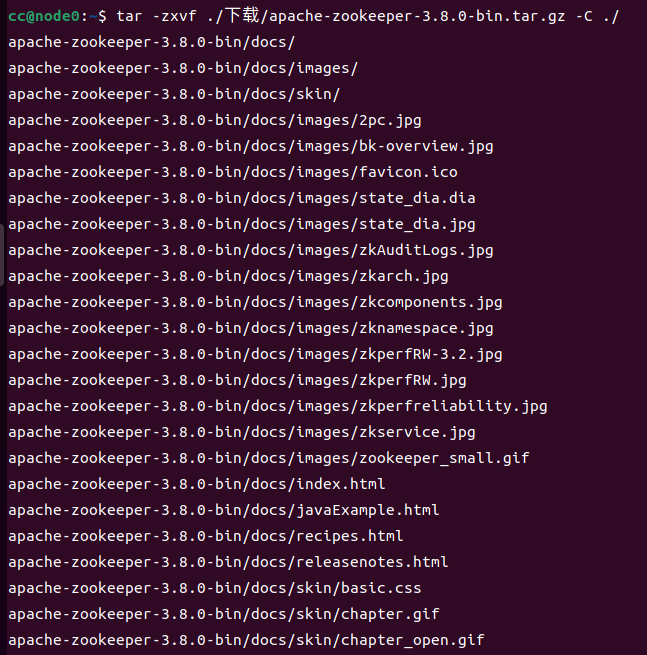
restrict 192.168.0.0 mask 255.255.0.0 nomodify



配置完成后“:wq”保存退出

解压zookeeper到当前文件夹：

tar -zxvf ./下载/apache-zookeeper-3.8.0-bin.tar.gz -C ./



把apache-zookeeper-3.8.0-bin改名为zookeeper：

mv apache-zookeeper-3.8.0-bin zookeeper



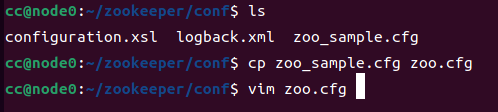
先cd到zookeeper里的conf目录下：

cd zookeeper/conf



复制zoo\_sample.cfg并命名为zoo.cfg：cp zoo\_sample.cfg zoo.cfg

然后vim编辑zoo.cfg文件：vim zoo.cfg



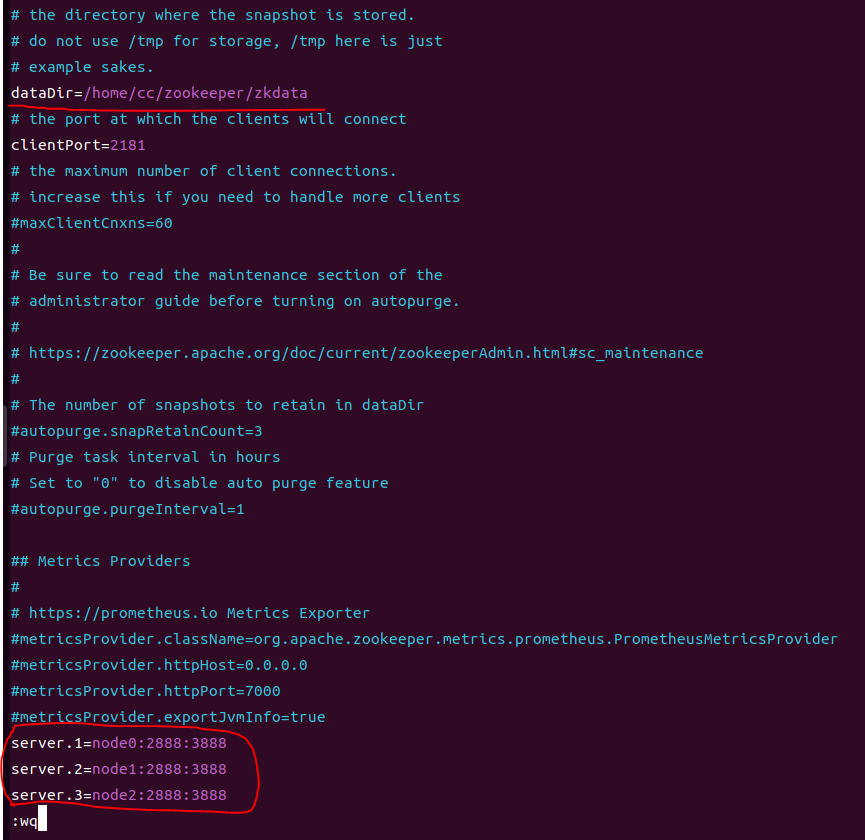
进入后自定义修改dataDir的目录位置，

并在文件末尾加入

server.1=node0:2888:3888

server.2=node1:2888:3888

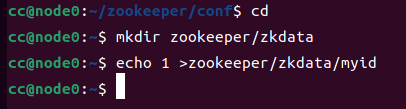
server.3=node2:2888:3888



然后“:wq”保存退出

cd回到当前用户主目录然后创建刚刚自定义的zkdata文件夹

并将配置中对应的节点编号保存到名为myid的文件中（现在配的节点是node0，对应了上图的，server.1=node0:2888:2888，所以myid内容为1）



修改环境配置文件：

vim /etc/profile



在下图位置添加环境变量：

export PATH=$PATH:/home/cc/zookeeper/bin



然后“:wq”保存退出

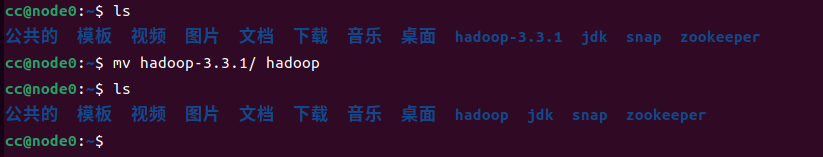
解压hadoop到当前目录：

tar -zxvf 下载/Hadoop-3.3.1.tar.gz -C ./



将hadoop-3.3.1改名为hadoop：

mv Hadoop-3.3.1/ hadoop



修改环境配置文件：

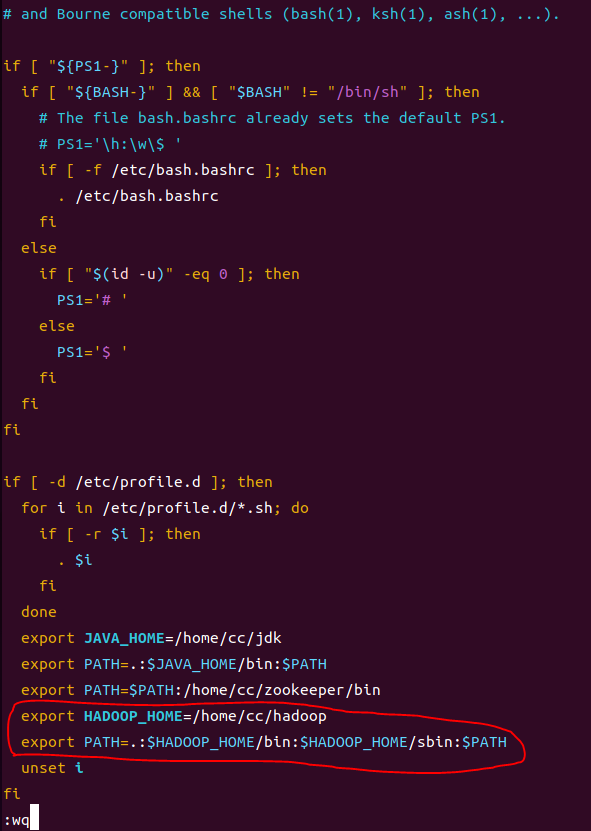
vim /etc/profile



在下图位置添加两条环境变量：

export HADOOP\_HOME=/home/cc/hadoop

export PATH=.:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin:$PATH



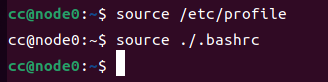
然后“:wq”保存退出

用source命令使Shell读入指定的Shell程序文件并依次执行文件中的所有语句：

source /etc/profile

因为我这里source后“cc@node0”变成白色了，本文档上面有修改过cc用户目录下的.bashrc文件，解决了变色问题，所以这里用source命令重新执行.bashrc配置文件即可：

source ./.bashrc



查看hadoop版本信息来验证配置成功：

hadoop version



cd到hadoop安装目录下的etc/hadoop目录：

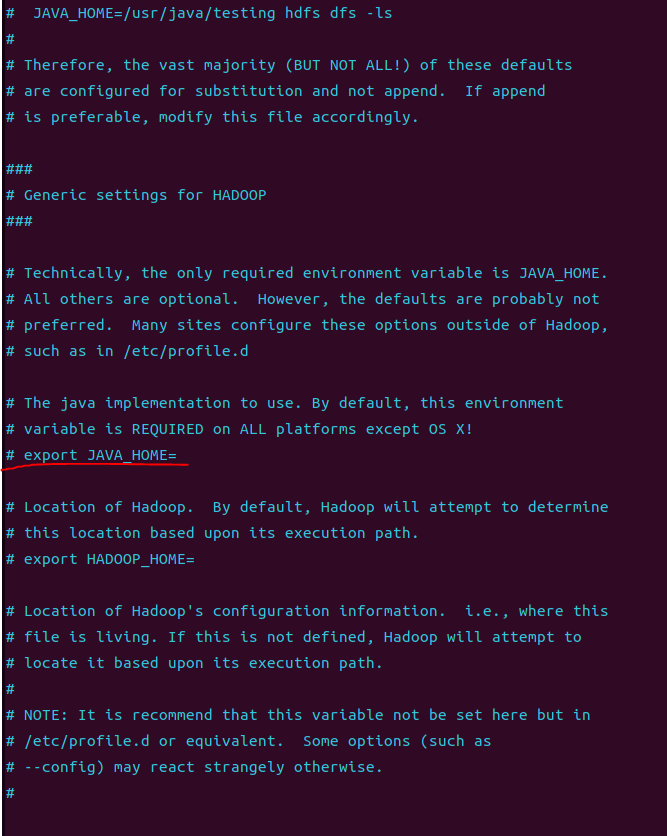
cd hadoop/etc/Hadoop

然后修改配置文件hadoop-env.sh：

vim hadoop-env.sh

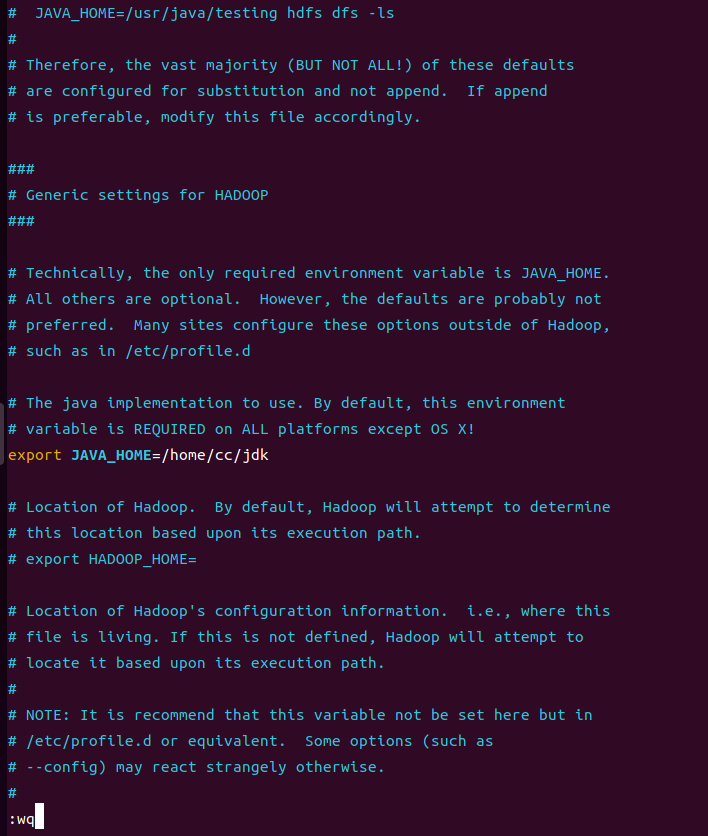


找到下图所示位置：



把“#export JAVA\_HOME=”修改为：

export JAVA\_HOME=/home/cc/jdk



然后“:wq”保存退出

修改hadoop的core-site.xml配置文件：

vim core-site.xml



修改内容为：

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://ns</value>

</property>

<property>

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

<value>/home/cc/hadoop/metadata</value>

</property>

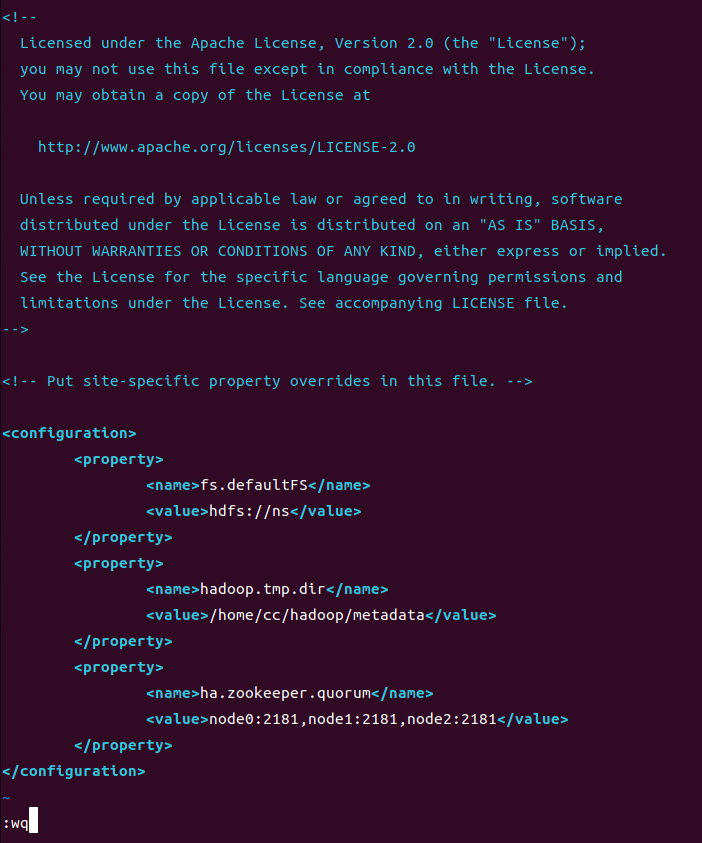
<property>

<name>ha.zookeeper.quorum</name>

<value>node0:2181,node1:2181,node2:2181</value>

</property>

</configuration>



然后“:wq”保存退出

修改hadoop的hdfs-site.xml配置文件：

vim hdfs-site.xml



进去后修改内容为：

<configuration>

<property>

<name>dfs.replication</name>

<value>3</value>

</property>

<property>

<name>dfs.nameservices</name>

<value>ns</value>

</property>

<property>

<name>dfs.ha.namenodes.ns</name>

<value>nn1,nn2</value>

</property>

<property>

<name>dfs.namenode.rpc-address.ns.nn1</name>

<value>node0:8020</value>

</property>

<property>

<name>dfs.namenode.http-address.ns.nn1</name>

<value>node0:50070</value>

</property>

<property>

<name>dfs.namenode.rpc-address.ns.nn2</name>

<value>node1:8020</value>

</property>

<property>

<name>dfs.namenode.http-address.ns.nn2</name>

<value>node1:50070</value>

</property>

<property>

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

<value>qjournal://node0:8485;node1:8485;node2:8485/ns</value>

</property>

<property>

<name>dfs.journalnode.edits.dir</name>

<value>/home/cc/hadoop/ha/journal</value>

</property>

<property>

<name>dfs.ha.automatic-failover.enabled</name>

<value>true</value>

</property>

<property>

<name>dfs.client.failover.proxy.provider.ns</name>

<value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>

</property>

<property>

<name>dfs.ha.fencing.methods</name>

<value>sshfence</value>

</property>

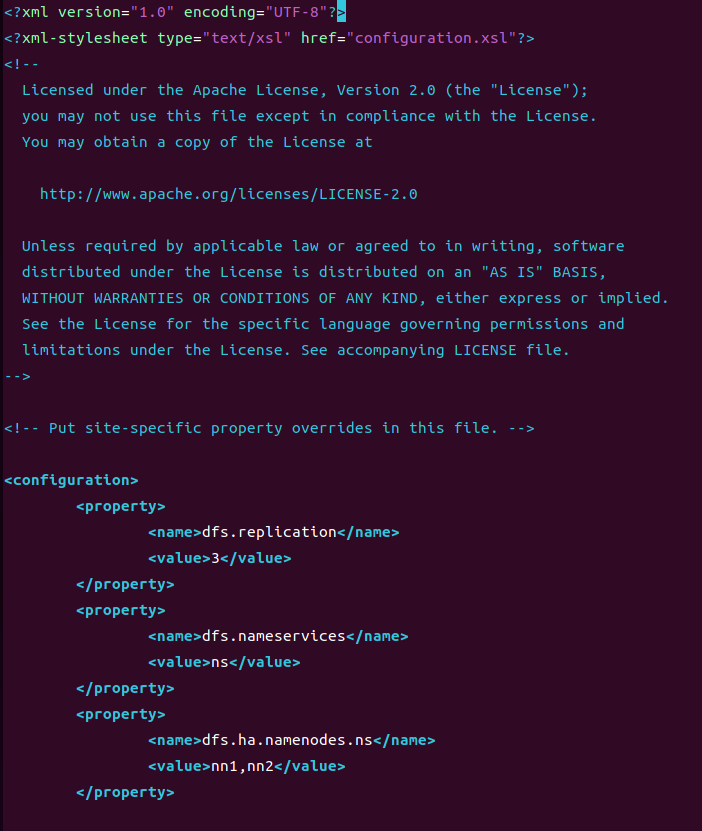
<property>

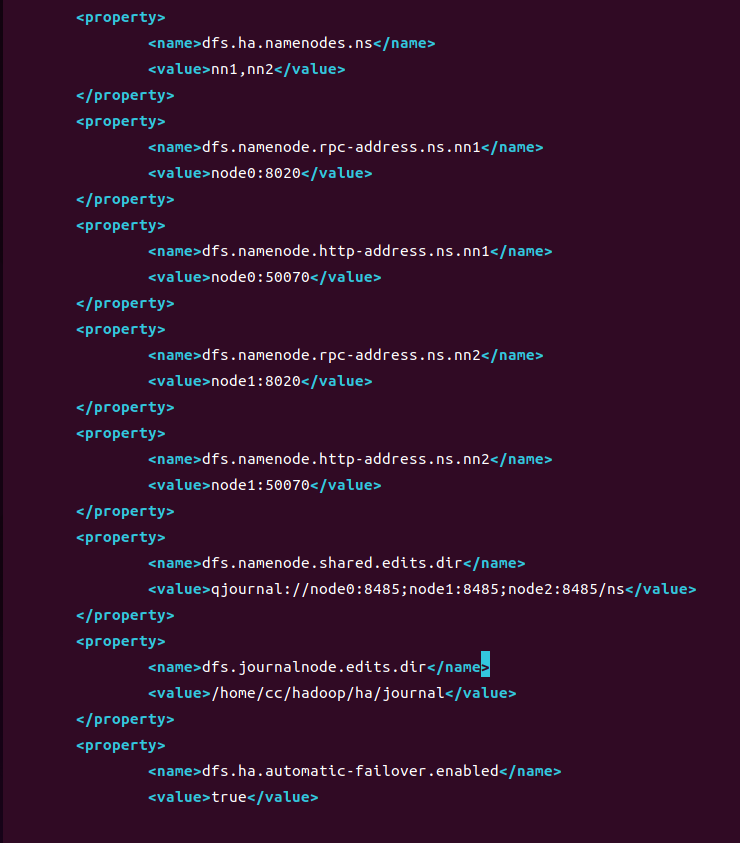
<name>dfs.ha.fencing.ssh.private-key-files</name>

<value>/home/cc/.ssh/id\_rsa</value>

</property>

</configuration>







然后“:wq”保存退出

修改hadoop的mapred-site.xml配置文件：

vim mapred-site.xml



进去后修改内容为：

<configuration>

<property>

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

<value>yarn</value>

</property>

<property>

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

<value>node0:10020</value>

</property>

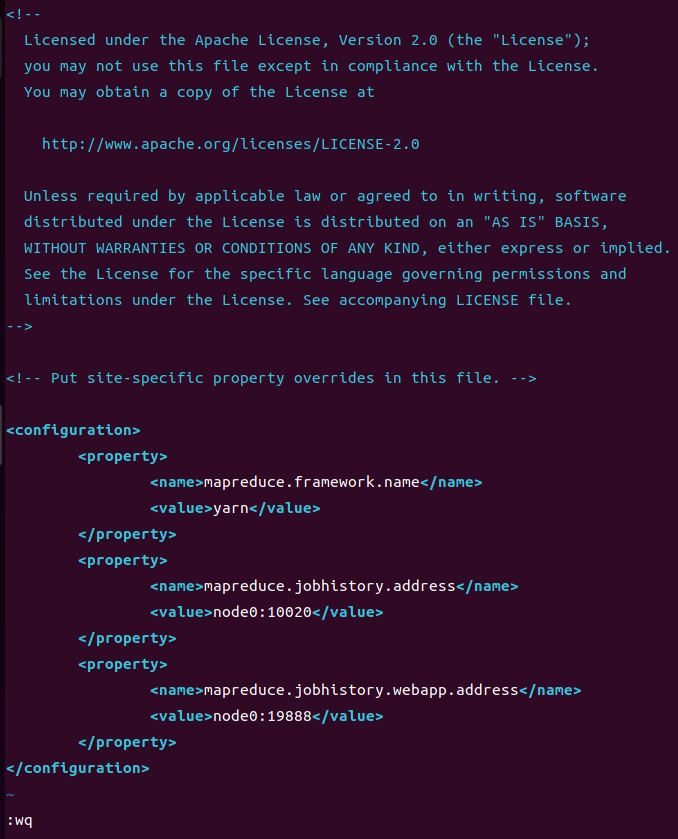
<property>

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

<value>node0:19888</value>

</property>

</configuration>



然后“:wq”保存退出

修改hadoop的yarn-site.xml配置文件：

vim yarn-site.xml



进去后修改内容为：

<configuration>

<property>

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

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.resourcemanager.ha.enabled</name>

<value>true</value>

</property>

<property>

<name>yarn.resourcemanager.cluster-id</name>

<value>RM\_CLUSTER</value>

</property>

<property>

<name>yarn.resourcemanager.ha.rm-ids</name>

<value>rm1,rm2</value>

</property>

<property>

<name>yarn.resourcemanager.hostname.rm1</name>

<value>node0</value>

</property>

<property>

<name>yarn.resourcemanager.hostname.rm2</name>

<value>node1</value>

</property>

<property>

<name>yarn.resourcemanager.recovery.enabled</name>

<value>true</value>

</property>

<property>

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

<value>node0:2181,node1:2181,node2:2181</value>

</property>

</configuration>





然后“:wq”保存退出

修改配置文件workers：

vim workers

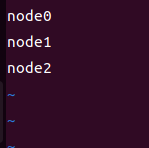


进去后删除“localhost”并修改为如下信息：

node0

node1

node2

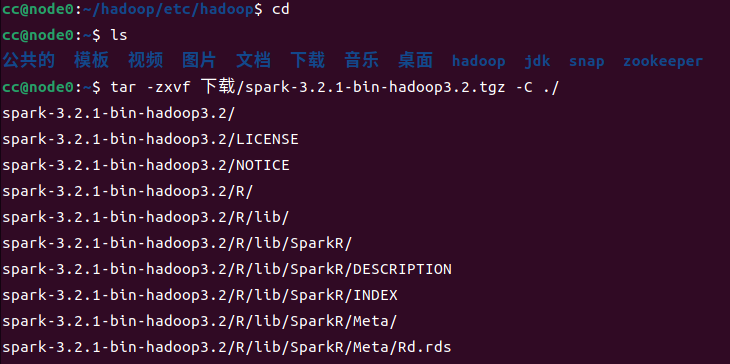


然后“:wq”保存退出

cd回到用户主目录，解压用户目录中下载文件夹中的spark到当前文件夹：

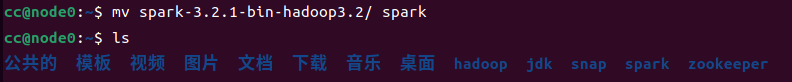
cd

tar -zxvf 下载/spark-3.2.1-bin-hadoop3.2.tgz -C ./



把spark-3.2.1-bin-hadoop3.2改名为spark：

mv spark-3.2.1-bin-hadoop3.2 spark

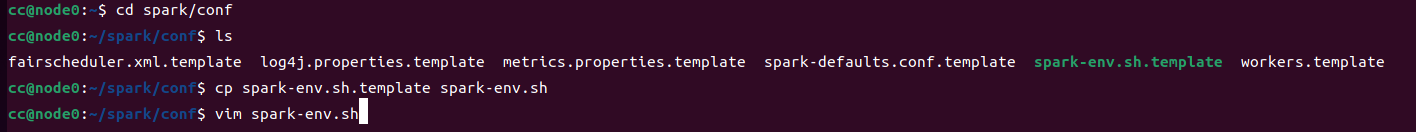


cd到spark的conf目录下，复制spark-env.sh.template文件并命名为spark-env.sh然后编辑spark-env.sh：

cd spark/conf

cp spark-env.sh.template spark-env.sh

vim spark-env.sh



进入后在最上面添加以下内容：

export JAVA\_HOME=/home/cc/jdk

export HADOOP\_CONF\_DIR=/home/cc/hadoop/etc/hadoop

export SPARK\_MASTER\_IP=node0

export SPARK\_MASTER\_PORT=7077

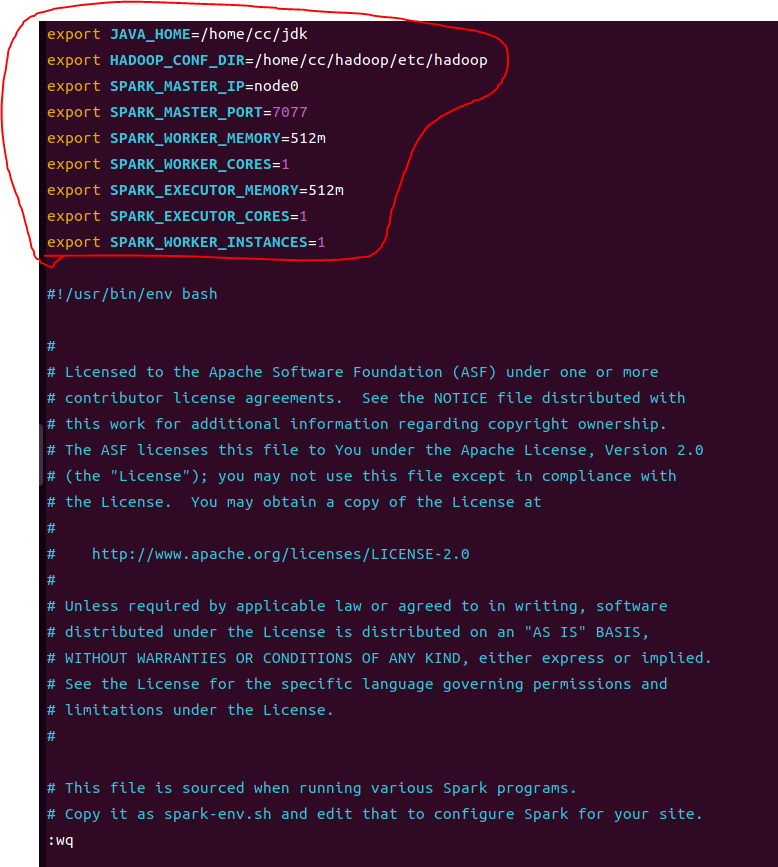
export SPARK\_WORKER\_MEMORY=512m

export SPARK\_WORKER\_CORES=1

export SPARK\_EXECUTOR\_MEMORY=512m

export SPARK\_EXECUTOR\_CORES=1

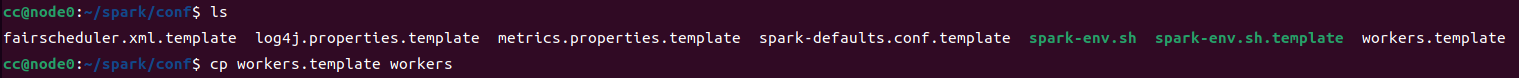
export SPARK\_WORKER\_INSTANCES=1



然后“:wq”保存退出

复制workers.template文件并命名为workers：

cp workers.template workers

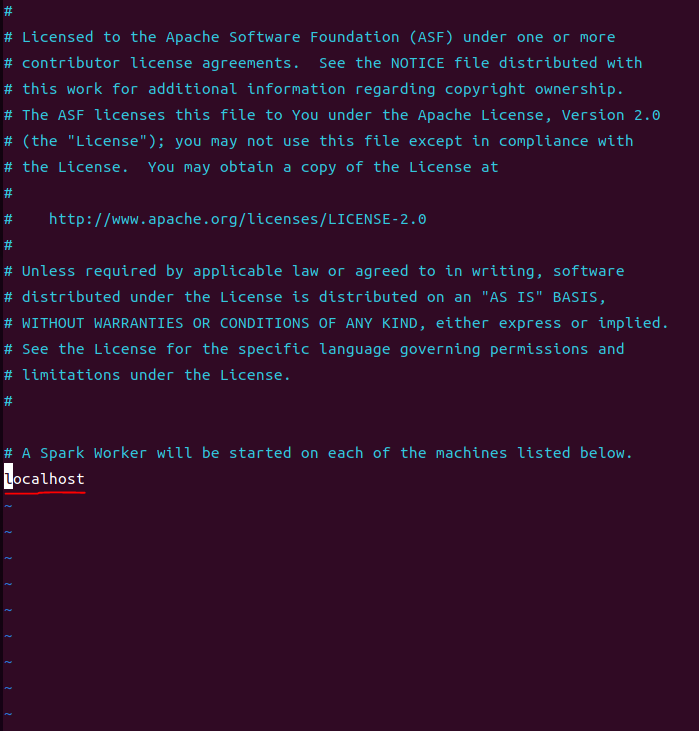


修改workers文件：

vim workers



删掉里面的localhost

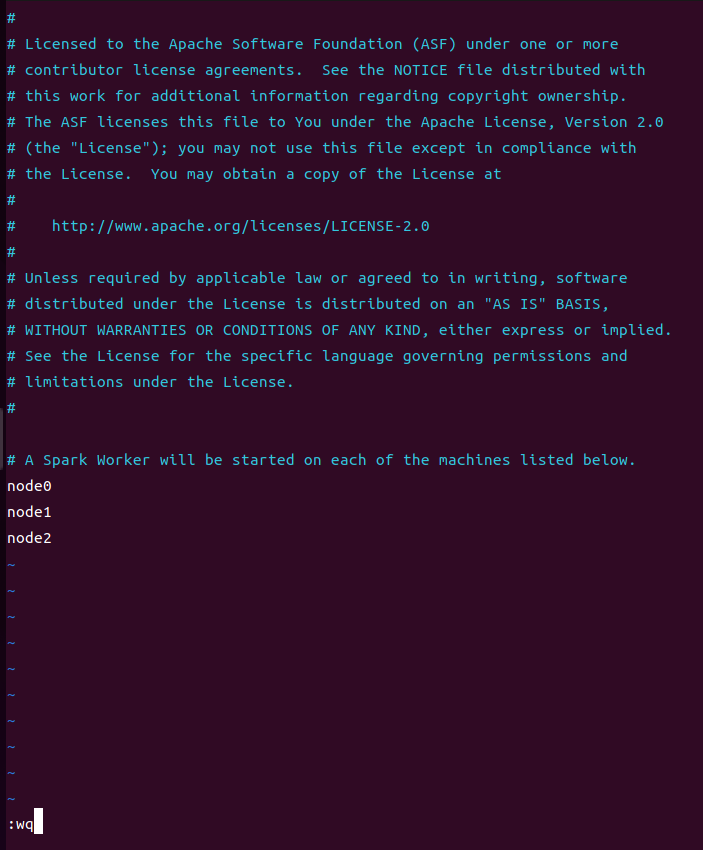


添加：

node0

node1

node2

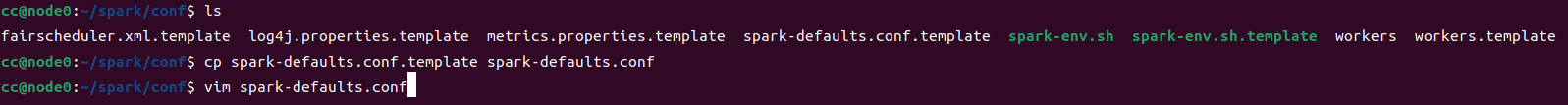


然后“:wq”保存退出

复制spark-defaults.conf.template文件并命名为spark-default.conf，然后编辑spark-default.conf：

cp spark-defaults.conf.template spark-default.conf

vim spark-default.conf







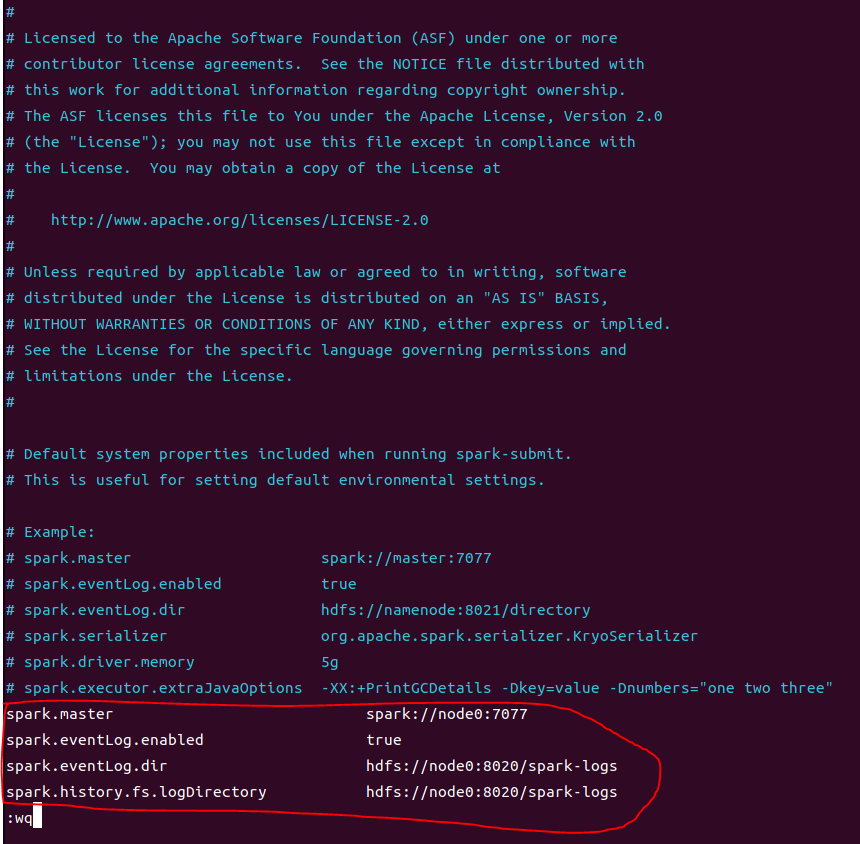
在末尾添加：

spark.master spark://node0:7077

spark.eventLog.enabled true

spark.eventLog.dir hdfs://node0:8020/spark-logs

spark.history.fs.logDirectory hdfs://node0:8020/spark-logs



然后“:wq”保存退出

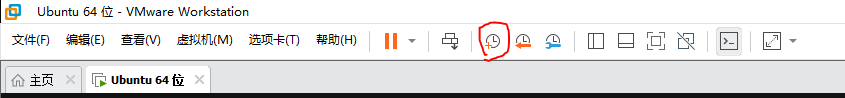
在tmp下创建spark-events文件夹：

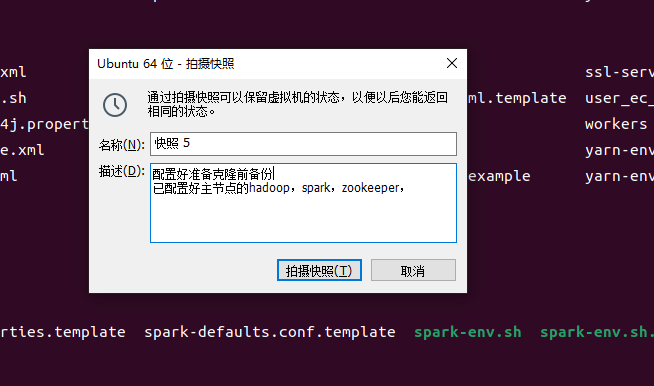
mkdir /tmp/spark-events

补充：如果没有这个文件夹，可能会令spark的history（start-history-server.sh命令）启动失败



准备克隆，先点击拍摄快照来备份：





关机命令：

init 0



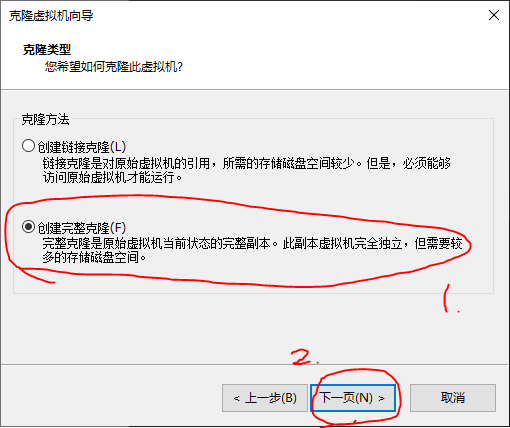
右键此处，点“管理”然后点“克隆”

补充：克隆机和原机的网卡的物理地址相同，会影响桥接网络，如果要使用桥接，记得先刷新一下克隆机的物理地址

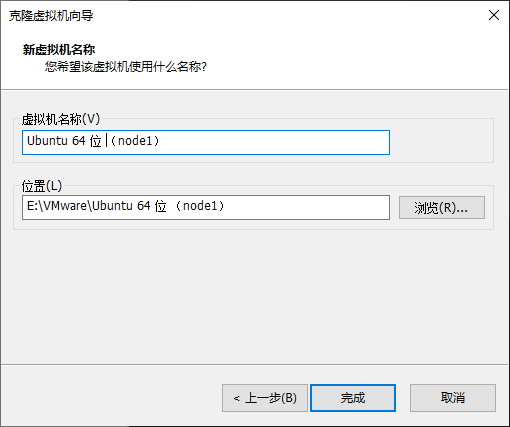




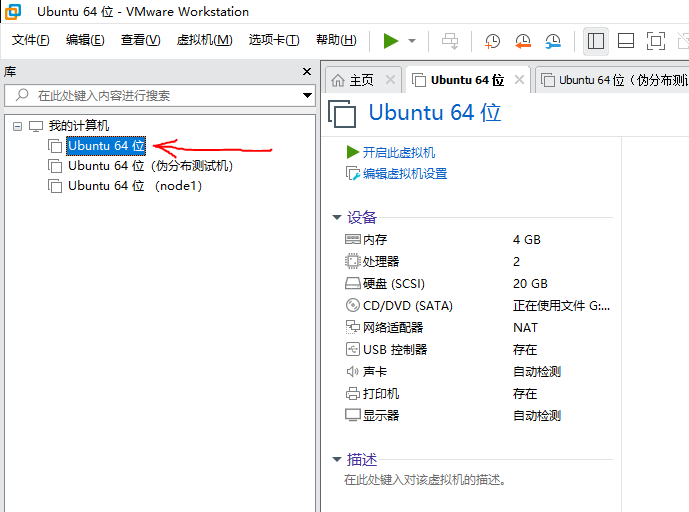


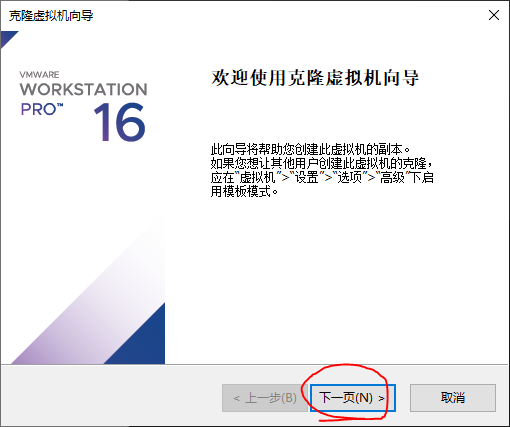


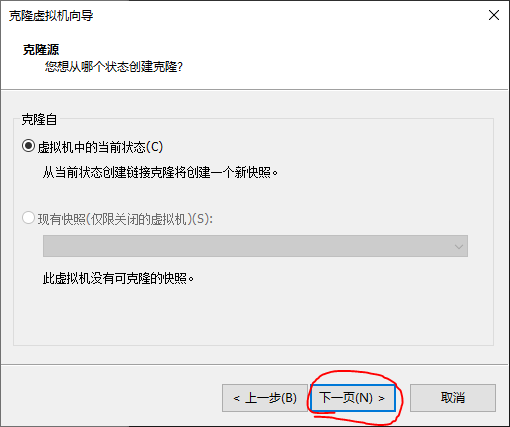
自定义虚拟机名称和位置后点完成：

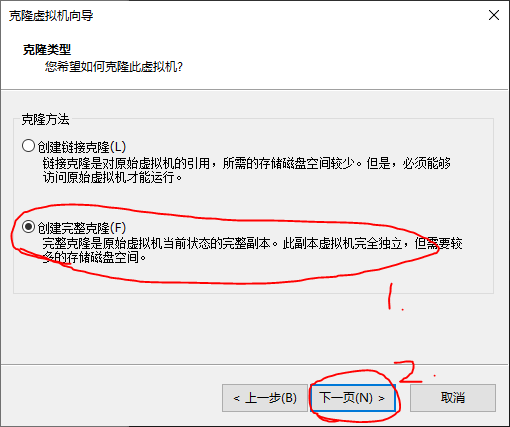


克隆完成后，再克隆一次：

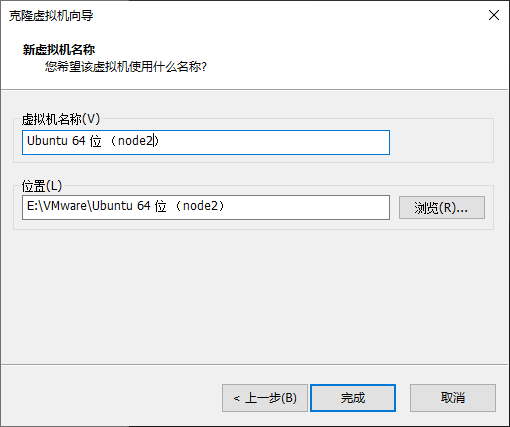








自定义虚拟机名称和位置后点完成：



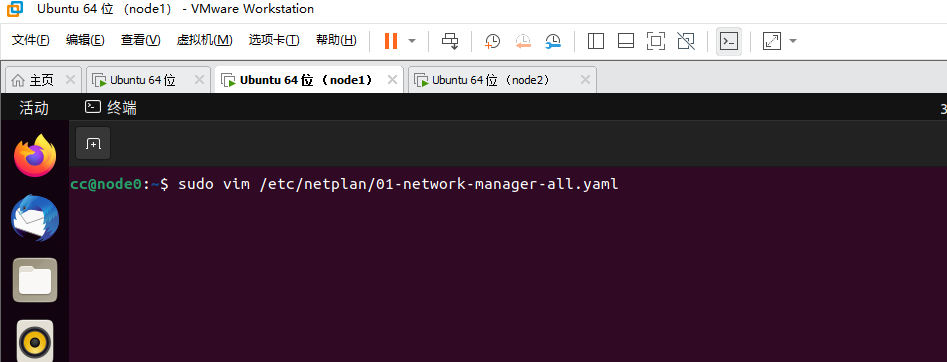
克隆完成后运行这三台虚拟机：



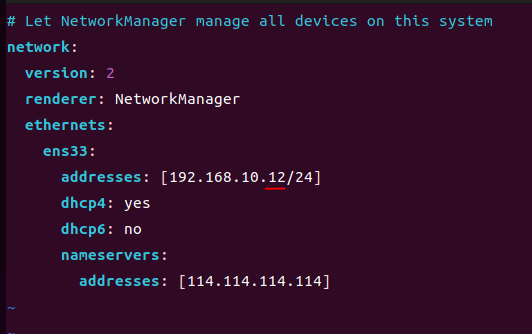
**---------------------------以下配置node1--------------------------------------**

**在node1上修改网卡：**

sudo vim /etc/netplan/01-network-manager-all.yaml



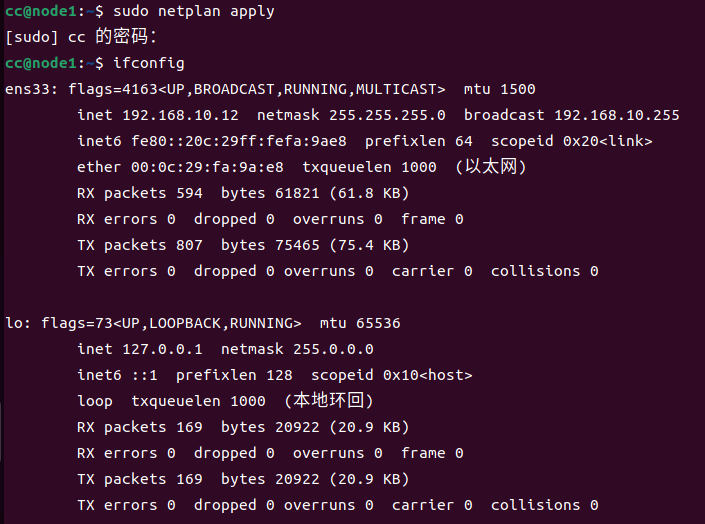
修改ip地址为192.168.10.12：



然后“:wq”保存退出

重启网络（这里截图是后来补上的，所以显示cc@node1，原来应该是cc@node0，因为还没改主机名）：

sudo netplan apply

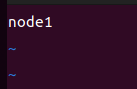


修改配置主机名：

sudo vim /etc/hostname



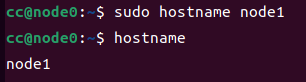
把node0改为node1



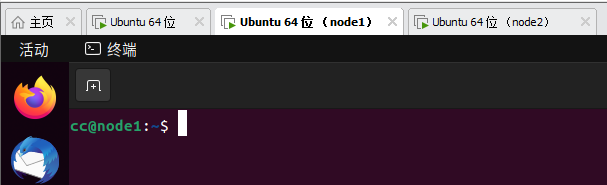
然后保存“:wq”保存退出

但hostname查看结果还是node0是正常的，修改hostname配置文件要重启才能生效，如果不想重启可用“sudo hostname node1”暂时修改主机名：





关闭当前终端新开一个终端，“cc@node0”变成“cc@node1”：

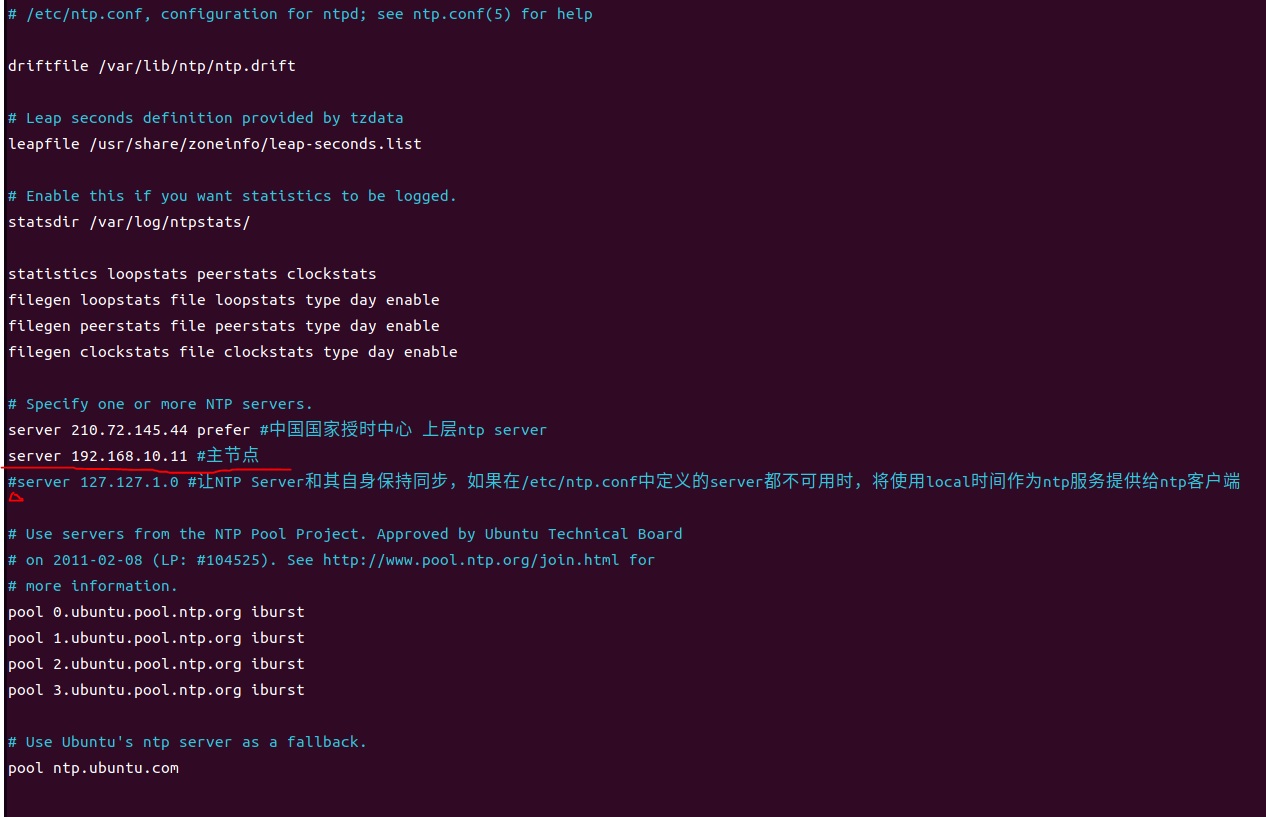


修改ntp配置文件：

sudo vim /etc/ntp.conf



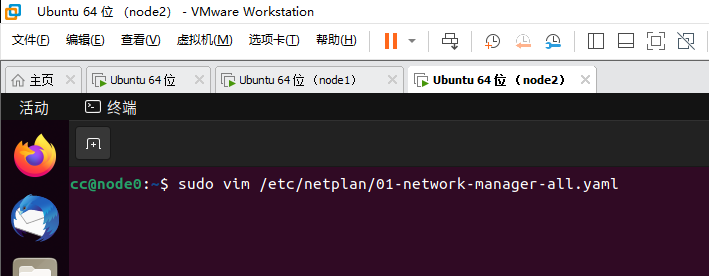
添加主节点为服务器，并注释掉“server 127.127.1.0”



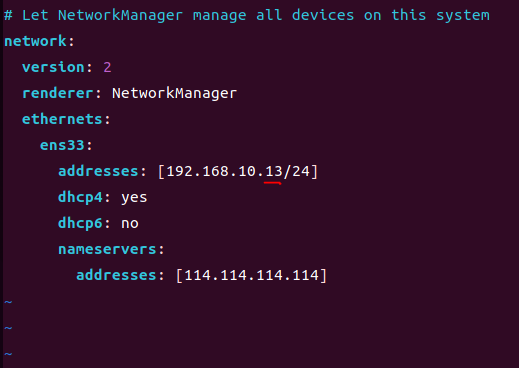
**--------------------------------以下配置node2---------------------------------**

**在node2上修改网卡：**

sudo vim /etc/netplan/01-network-manager-all.yaml



修改ip地址为192.168.10.13：



然后“:wq”保存退出

重启网络（这里截图是后来补上的，所以显示cc@node2，原来应该是cc@node0，因为还没改主机名）：

sudo netplan apply

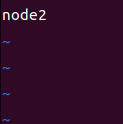


修改配置主机名：

sudo vim /etc/hostname



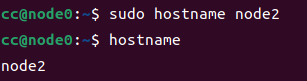
把node0改为node2



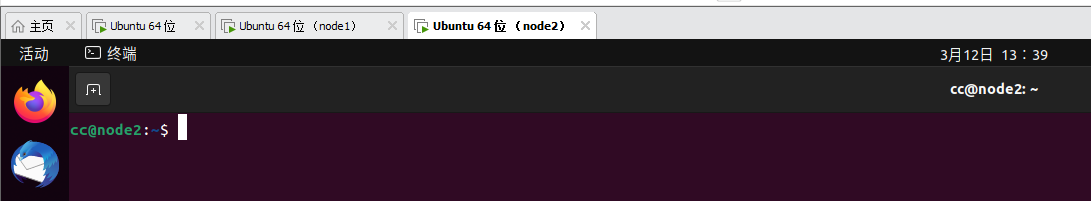
然后保存“:wq”保存退出

但hostname查看结果还是node0是正常的，那是因为修改hostname配置文件要重启才能生效，如果不想重启可用“sudo hostname node2”暂时修改主机名：





关闭当前终端新开一个终端，“cc@node0”变成“cc@node2”：

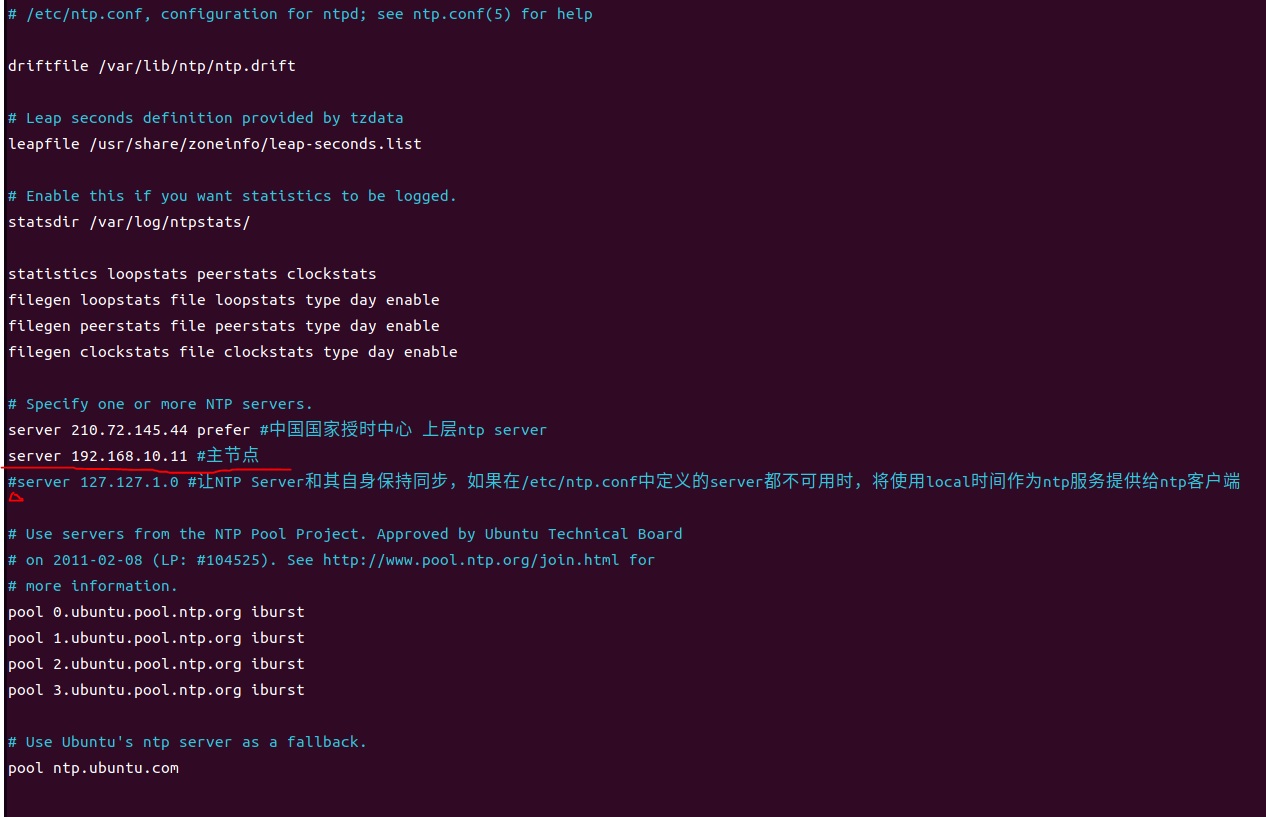


修改ntp配置文件：

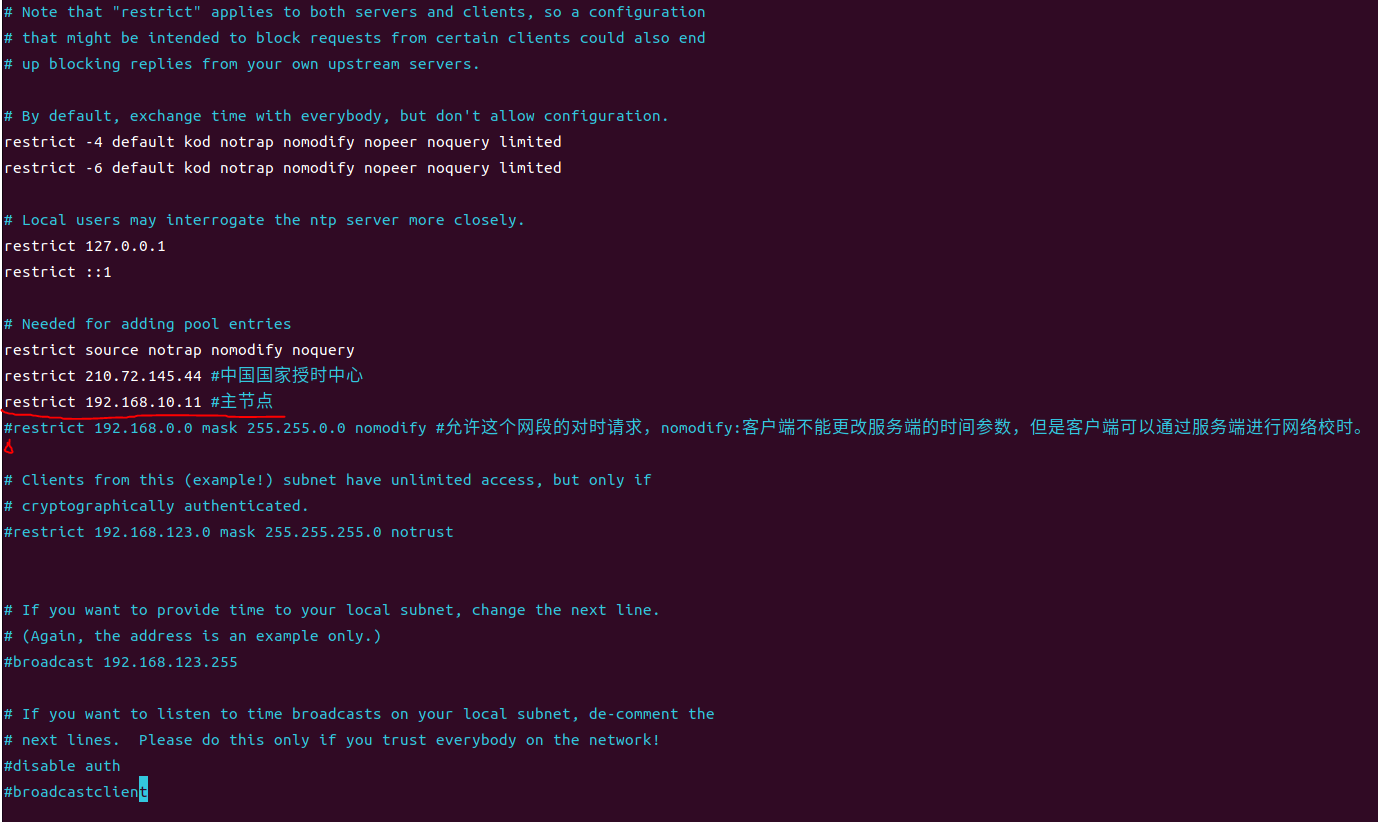
sudo vim /etc/ntp.conf



添加主节点为服务器，并注释掉“server 127.127.1.0”



来到下面部分，添加“restrict 192.168.10.11”，注释掉“restrict 192.168.0.0 mask 255.255.255.0 nomodify”：



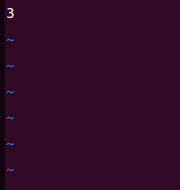
然后“:wq”保存退出

修改zookeeper的myid文件：

vim zookeeper/zkdata/myid



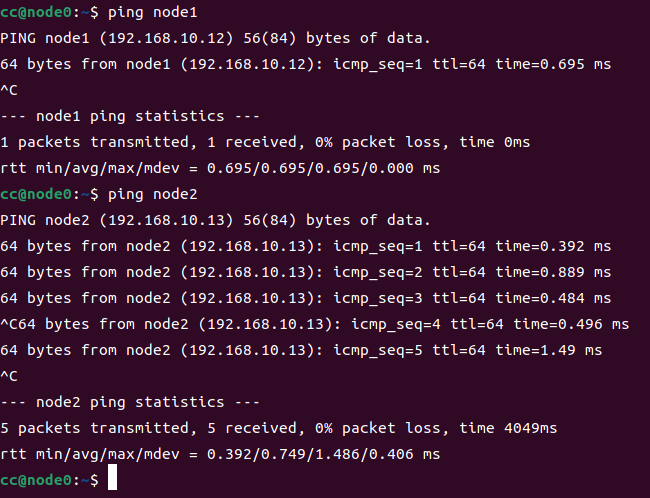
进入后把“1”改成“3”：



然后“:wq”保存退出

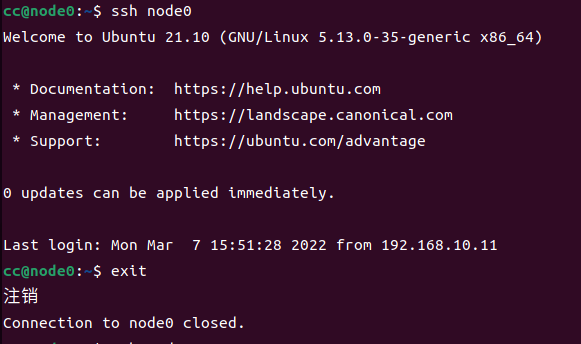
**------------------------------------回到node0---------------------------------**

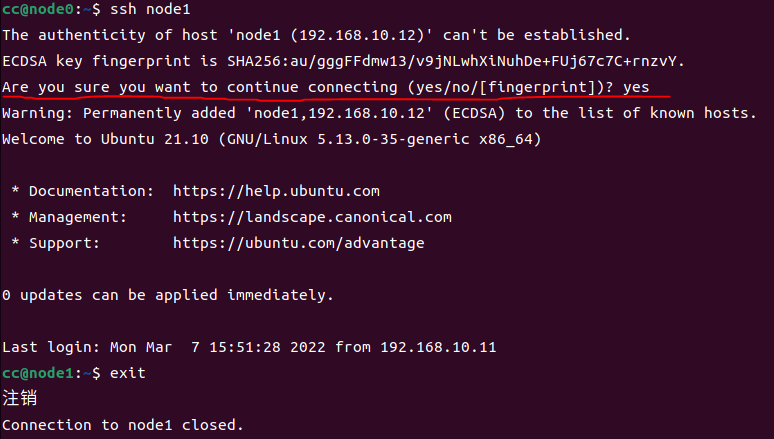
尝试与node1和node2的网络通信：

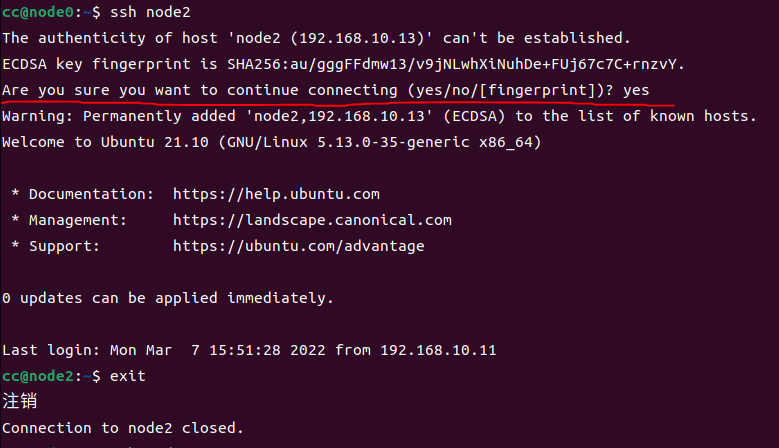


尝试用ssh登录node0，node1，node2（第一次ssh登录会询问，输入“yes”即可，以后再用shh不会再询问）

补充：因为是克隆机，所以ssh的公钥密钥等文件和主机一样，所以不用重新配置ssh密钥都可以实现互相远程登录，但3台机公钥密钥都一样可能会存在一定的安全问题，可通过刷新每台机器的密钥，重新配置ssh解决。

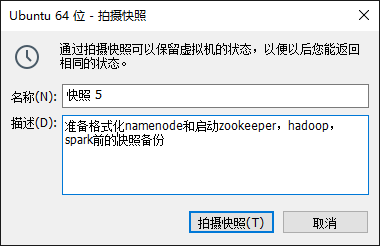


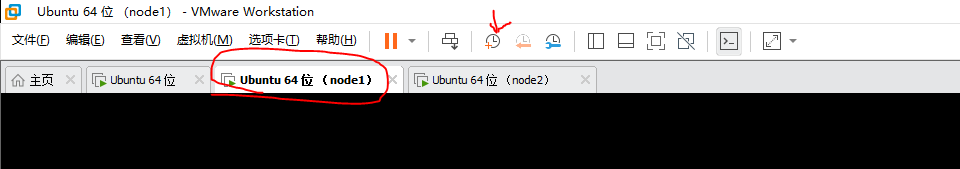


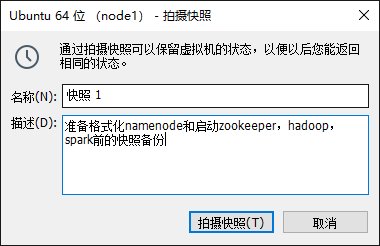


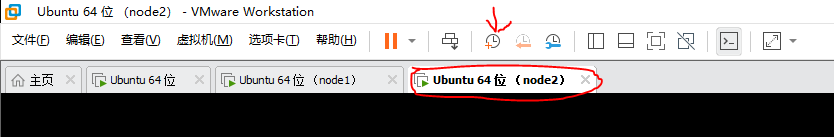
准备格式化namenode和启动zookeeper，hadoop，spark前每个节点都拍摄快照用以备份：

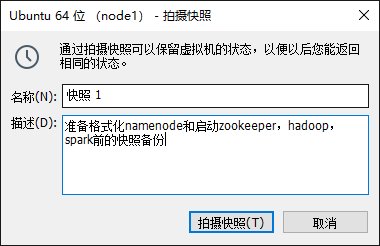






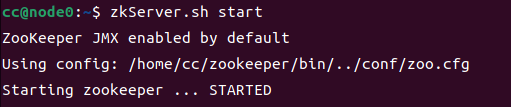






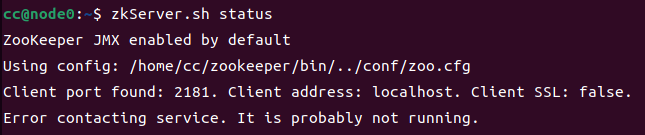
启动zookeeper：

zkServer.sh start



查看zookeeper状态发现错误是正常现象，因为还有两个节点上的zookeeper还未启动：

zkServer.sh status

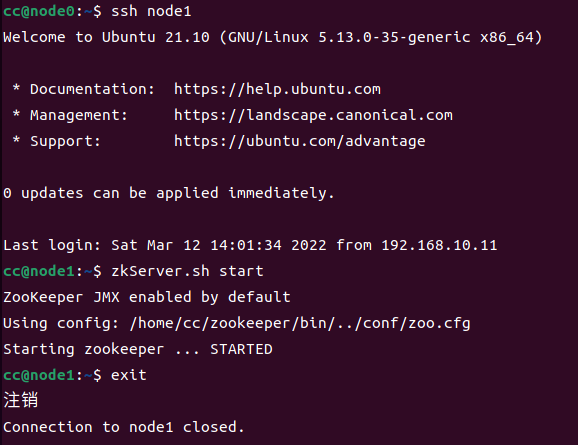


用ssh远程登录node1，然后启动zookeeper服务后exit退出远程登录回到node0：

ssh node1

zkServer.sh start

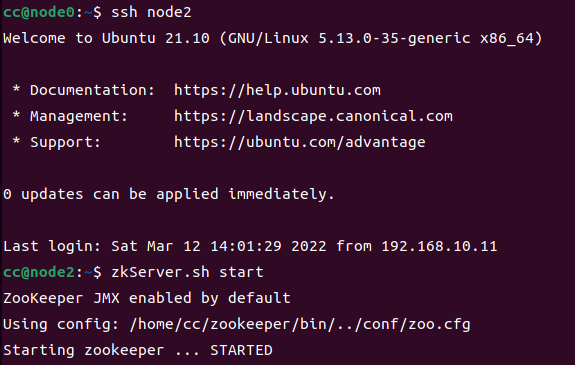
exit



用ssh远程登录node2，然后启动zookeeper服务：

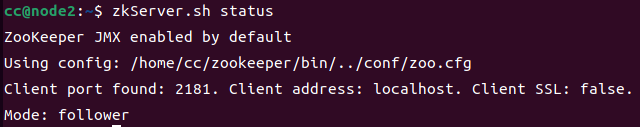
ssh node2

zkServer.sh start



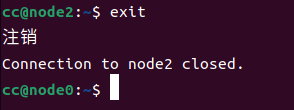
因为此时每个节点的zookeeper都已启动所以可以查看zookeeper服务的状态，出现“Mode：follower”或“Mode：leader”都表示正常运行：

zkServer.sh status



exit退出远程登录回到node0：

exit

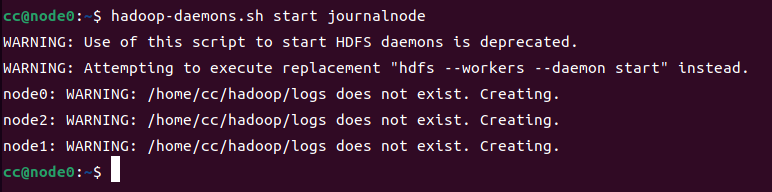


namenode要向journalnode写入日志

启动journalnode：

hadoop-daemons.sh start journalnode

注意：Attempting to execute replacement “hdfs --workers --daemon start” instead（旧命令已经过时了，建议用hdfs --workers --daemon start” instead来代替hadoop-daemons.sh start journalnode命令）



jps查看node0的进程后远程登录各个节点，检查每个节点的进程：

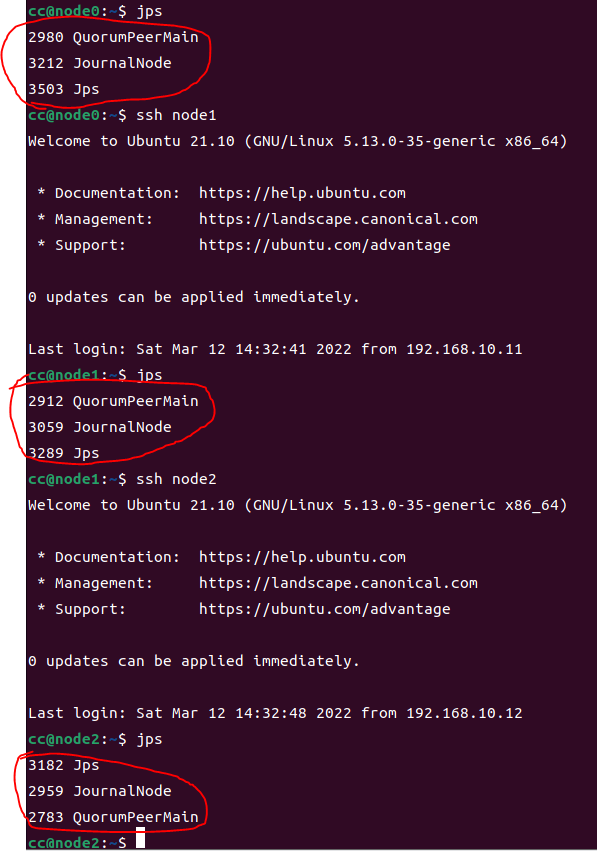
jps

ssh node1

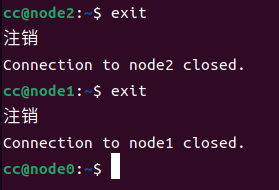
jps

ssh node2

jps

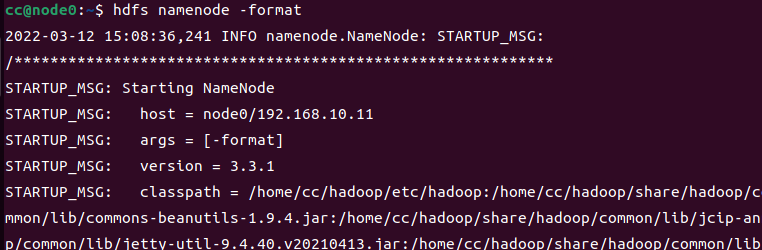


用exit回到node0：

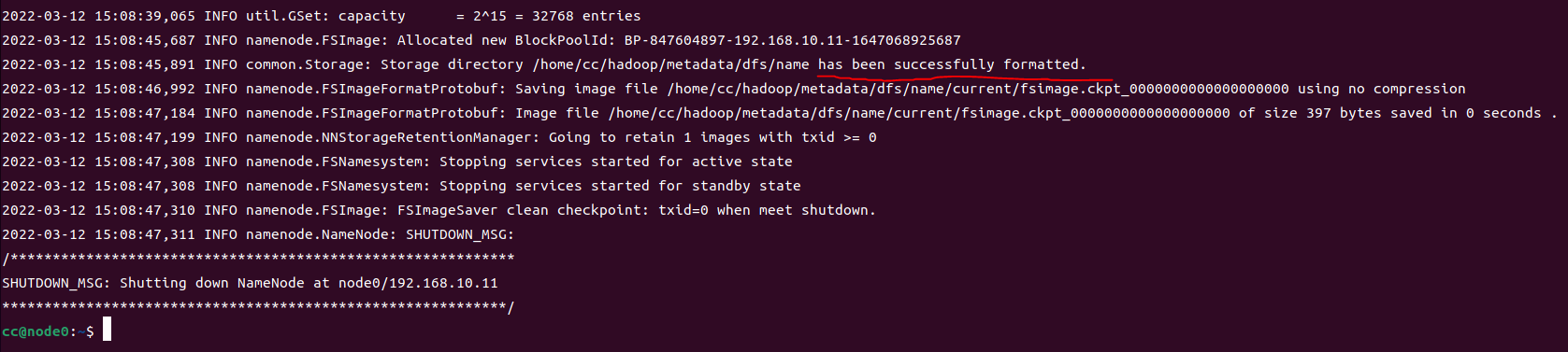


格式化namenode：

hdfs namenode -format



格式化成功：



格式化ZKFC：

hdfs zkfc -formatZK



补充：启动前需先将指定hadoop数据的临时存放目录（本文档的hadoop的配置文件core-site.xml中的metadata目录）用“scp -r”命令拷贝到第二个namenode节点，否则第二个namenode节点会没有namenode进程，日志会说namenode没有name文件（即namenode没有格式化）。

把node0的hadoop下的metadata目录用“scp -r”命令复制发送到node1的cc用户主目录的hadoop文件夹中：

scp -r hadoop/metadata cc@node1:/home/cc/hadoop



启动hadoop：

start-dfs.sh

start-yarn.sh



然后用jps和ssh查看各个节点的进程：

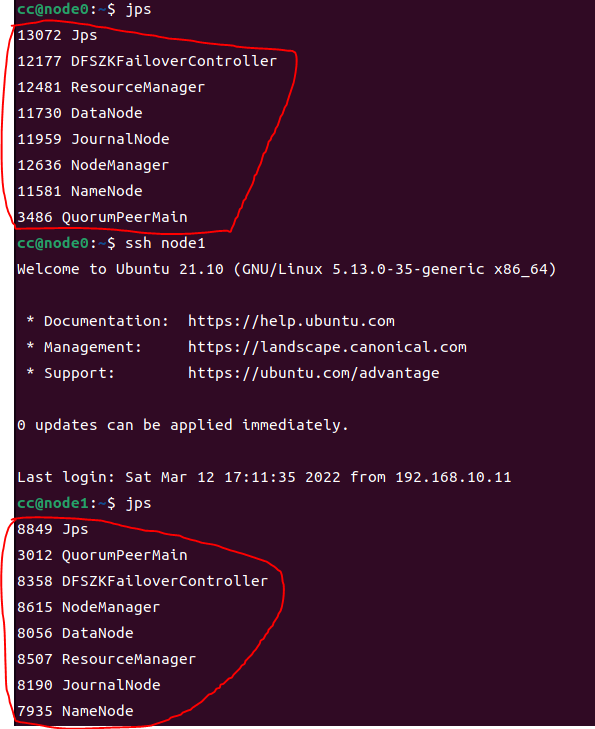
jps

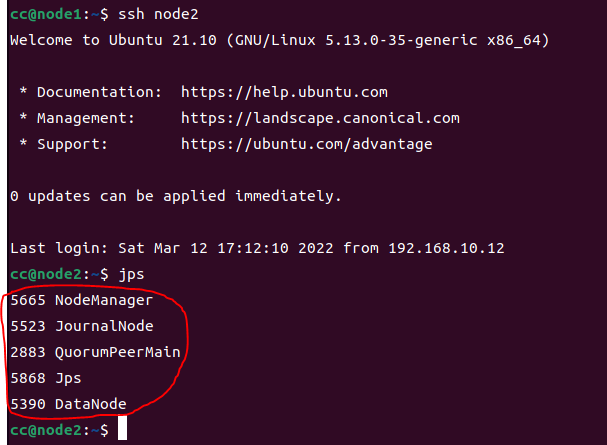
ssh node1

jps

ssh node2

jps





修改开放各个节点的hadoop的metadata文件夹的权限：

chmod -R a+w hadoop/metadata

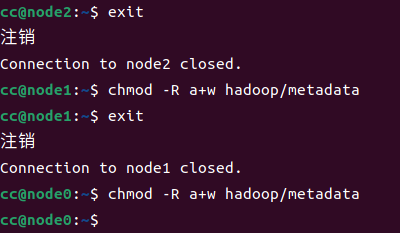
exit

chmod -R a+w hadoop/metadata

exit

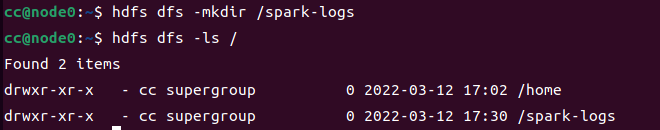
chmod -R a+w hadoop/metadata





在hdfs上创建spark-logs目录：

hdfs dfs -mkdir /spark-logs



cd到spark的sbin目录下：

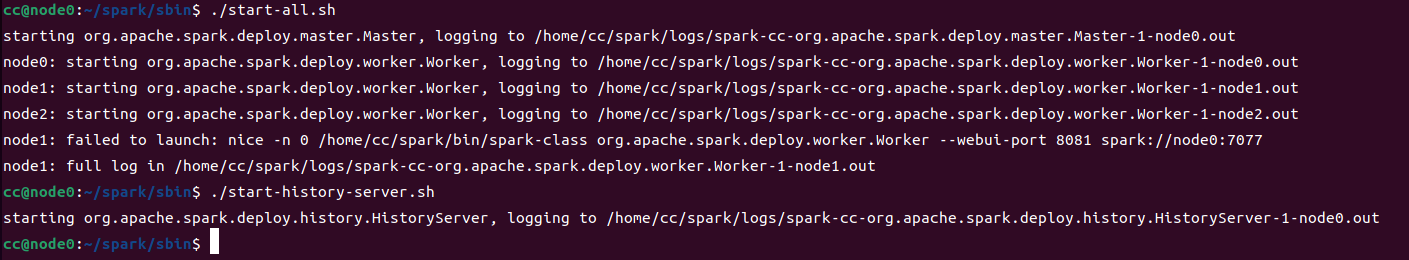
cd spark/sbin



启动spark集群：

./start-all.sh

./start-history-server.sh



查看各个节点的jps进程：

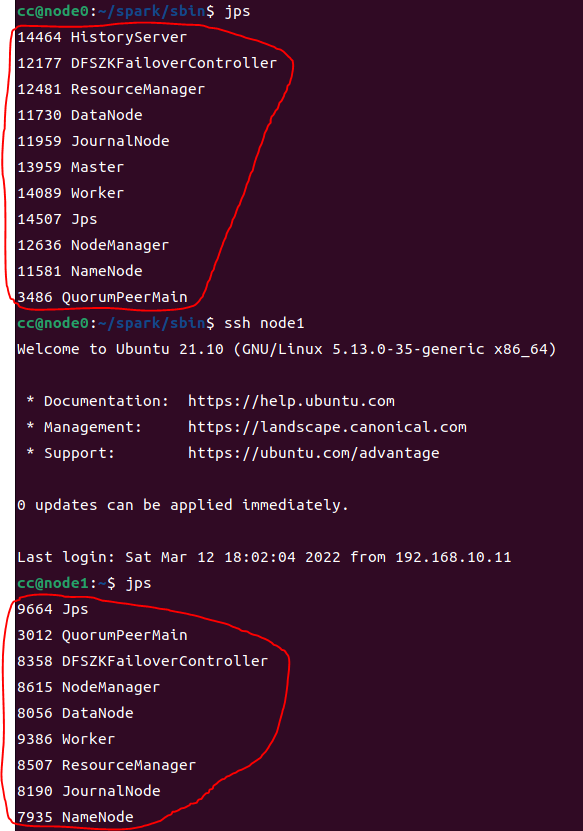
jps

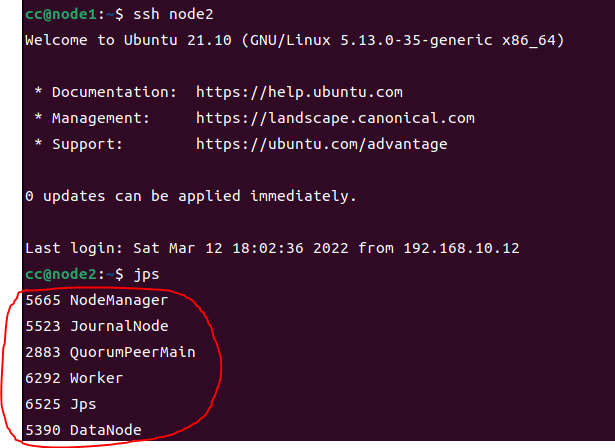
ssh node1

jps

ssh node2

jps

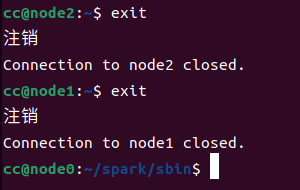




用exit回到node0：

exit

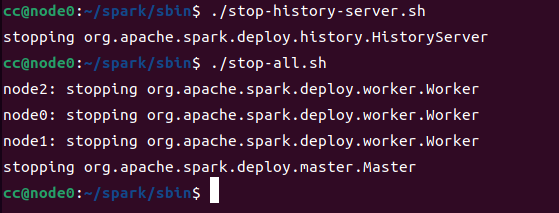
exit



关掉spark：

./stop-history-server.sh

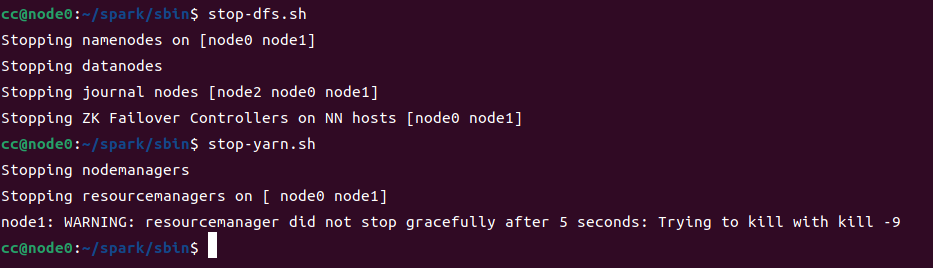
./stop-all.sh



关掉hadoop：

stop-dfs.sh

stop-yarn.sh



关闭zookeeper：

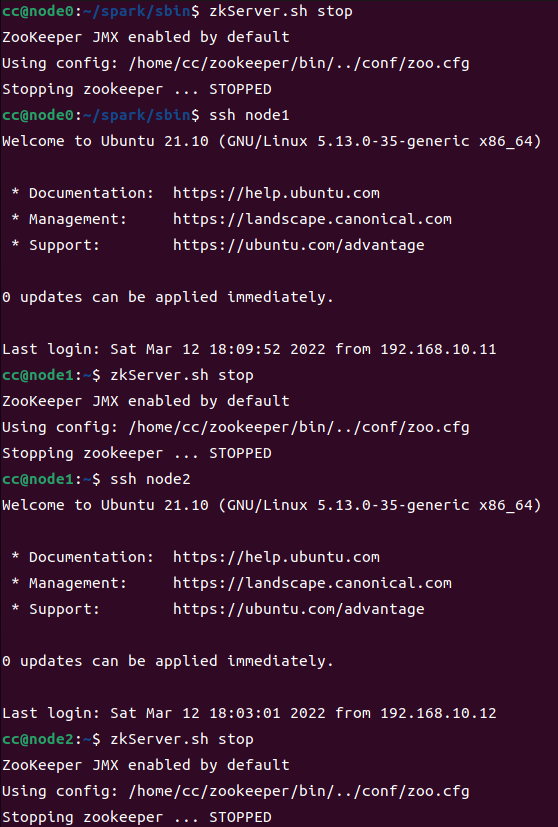
zkServer.sh stop

ssh node1

zkServer.sh stop

ssh node2

zkServer.sh stop



重复jps后exit检查是否关闭干净同时回到了node0：

jps

exit

jps

exit

jps

