1、修改主机名



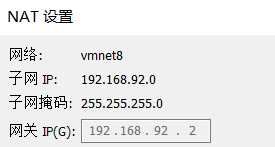




2、配置静态文件

打开虚拟网络编辑器，查看VMnet8的设置，根据设置中的默认网关、子网掩码设置静态ip，使其在同一网段。





（1）vi /etc/sysconfig/network-scripts/ifcfg-ens33

将最后一项ONBOOT=no改成 ONBOOT=yes

将BOOTPROTO=dhcp 改成 BOOTPROTO=static

增加：

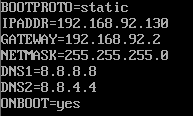
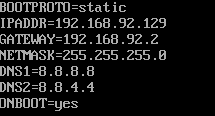
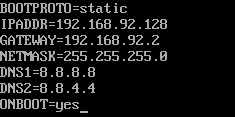
IPADDR=xxx.xxx.xxx.xxx

GATEWAY=192.168.92.2 #默认网关

NETMASK=255.255.255.0 #子网掩码

DNS1=8.8.8.8 #DNS1 配置

DNS2=8.8.4.4 #DNS2 配置



（2） systemctl status/start/stop/restart network.service

查看/开启/关闭/重启网络服务



3、配置host文件

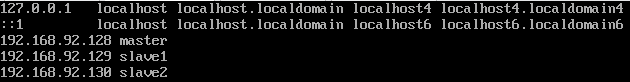
vi /etc/hosts

增加：(ip 计算机名)

192.168.92.128 master

192.168.92.129 slave1

192.168.92.130 slave2



4、关闭防火墙

将三个虚拟机的防火墙都关闭

查看防火墙状态：systemctl status firewalld.service

或者systemctl status firewalld

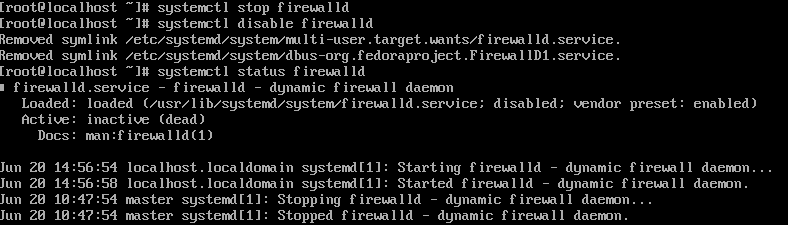
关闭防火墙

暂时关闭，重启系统后还会启动：

systemctl stop firewalld.service

启动关闭(enable/disable)：

systemctl disable firewalld.service



5、时间同步

hadoop集群对时间要求高，集群内同步。master做时间服务器，slave做客户端，与服务器时间保持一致

1）下载安装ntp(局域网内ntp时间同步(maser和几个slave都要安装ntp)):

yum install -y ntp

2）配置ntp(master作为服务器，修改ntp配置):

只在master虚拟机上配置ntp.conf

vi /etc/ntp.conf

在最后添加下面内容：

server 127.127.1.0 #local clock

fudge 127.127.1.0 stratum 10 #stratum 0~15



3） 重新启动ntp服务：systemctl restart ntpd.service



4）slave1和slave2同步时间：ntpdate master

6、配置ssh免密登陆

注：ssh免密登陆原理

在A端以 ssh-keygen 命令 生成一对公钥私钥，私钥存放在A端，公钥上传到B、C端。这样，每次从A端登陆到B、C端时，A端可凭借着私钥让B、C端认证自己。

1）三台虚拟机同时生成公钥、私钥

ssh-keygen -t rsa -f 文件名

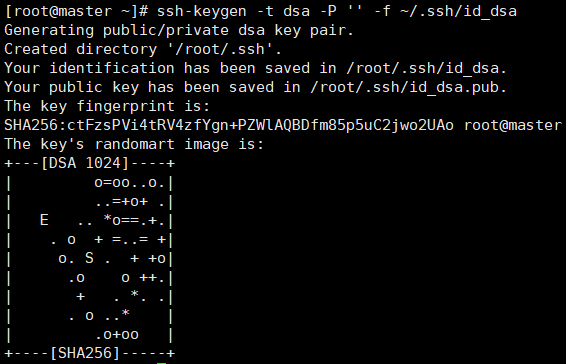
-t 类型 dsa 数字签名

生成两个密钥文件，一个是公钥，一个是私钥

命令:

ssh-keygen -t dsa -f ~/.ssh/id\_dsa

ssh-keygen -t dsa -P '' -f ~/.ssh/id\_dsa #生成公钥、私钥文件



(此时用ls .ssh命令可看到id\_dsa、 id\_dsa.pub两个私钥、公钥文件)



2）master节点：

将公钥文件复制到authorized\_keys并将文件权限改为600：

cat .ssh/id\_dsa.pub >> .ssh/authorized\_keys

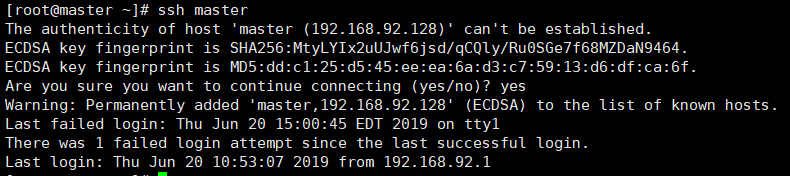
chmod 600 .ssh/authorized\_keys

ssh连接自己（ssh 内回环）：

ssh master（虚拟机名）

第一次需要输入验证:yes

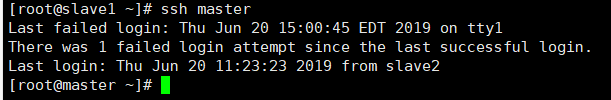
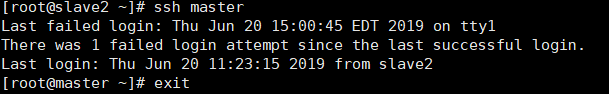
退出ssh连接：exit



3）将slave1和slave2上的公钥文件都合并到master的authorized\_keys文件中，使slave1和slave2对master 能够ssh免密登陆:

ssh root@slave1 cat /root/.ssh/id\_dsa.pub >> .ssh/authorized\_keys

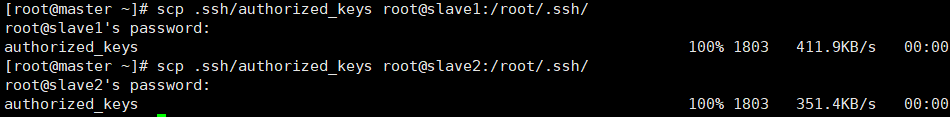
ssh root@slave2 cat /root/.ssh/id\_dsa.pub >> .ssh/authorized\_keys

4）将authorized\_keys复制到其他服务器的.ssh目录下，使master对slave1和slave2能够ssh免密登陆：

scp .ssh/authorized\_keys root@slave1:/root/.ssh/

scp .ssh/authorized\_keys root@slave2:/root/.ssh/







5）修改slave1和slave2的.ssh/authorized\_keys文件权限：

chmod 600 .ssh/authorized\_keys

7、安装JDK(三台虚拟机都需要安装)

1）创建目录：

mkdir /opt/soft

mkdir /user/java

2）使用xftp工具，把 jdk-8u171-linux-x64.tar.gz 文件拷贝到虚拟机的 /opt/soft 目录下

3）解压就是安装的第一步，第二步配置环境变量

tar -xzvf /opt/soft/jdk-8u171-linux-x64.tar.gz -C /usr/java

vi /etc/profile

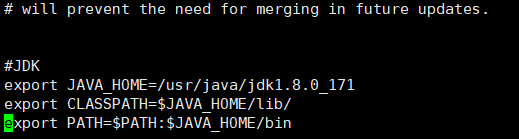
(在下面添加以下几条内容)

#JDK

export JAVA\_HOME=/usr/java/jdk1.8.0\_171

export CLASSPATH=$JAVA\_HOME/lib/

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



4）使环境变量生效：source /etc/profile

8、zookeeper安装和配置(先在master上操作，之后会有复制操作)

1）把电脑中的zookeeper软件上传到虚拟机master中

使用xftp工具，把 zookeeper-3.4.10.tar.gz 文件拷贝到虚拟机的 /opt/soft 目录下

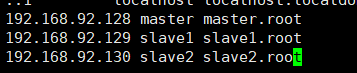
2）修改master主机名和ip地址的映射

vi /etc/hosts

192.168.121.128 master master.root

192.168.121.129 slave1 slave1.root

192.168.121.130 slave2 slave2.root



3）先在master这个虚拟机上安装zookeeper，安装目录：/usr/zookeeper

mkdir /usr/zookeeper

tar -xzvf /opt/soft/zookeeper-3.4.10.tar.gz -C /usr/zookeeper

4）配置处理：

zoo.cfg文件设置

cd /usr/zookeeper/zookeeper-3.4.10/conf

cp zoo\_sample.cfg zoo.cfg

vi zoo.cfg

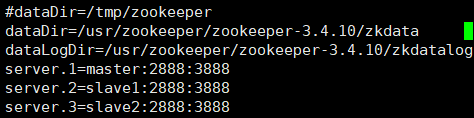
(以下为修改并增加的内容)

dataDir=/usr/zookeeper/zookeeper-3.4.10/zkdata #数据存储目录

dataLogDir=/usr/zookeeper/zookeeper-3.4.10/zkdatalog

server.1=master:2888:3888

server.2=slave1:2888:3888

server.3=slave2:2888:3888 

mkdir /usr/zookeeper/zookeeper-3.4.10/zkdata

mkdir /usr/zookeeper/zookeeper-3.4.10/zkdatalog

在/usr/zookeeper/zookeeper-3.4.10/zkdata中建立一个文件myid,文件保存一个数字：必须与zoo.cfg中的server.x=这台机器名:2888:3888一样

master：1

slave1：2

slave2：3

//server.x=xxxx:dddd:dddd

vi /usr/zookeeper/zookeeper-3.4.10/zkdata/myid

5） slave1和slave2也要修改主机名和ip地址的映射

a.在master上直接复制：

scp /etc/hosts root@slave1:/etc/hosts



scp /etc/hosts root@slave2:/etc/hosts



slave1和slave2也要安装配置zookeeper

b.复制master中的zookeeper目录到slave1和slave2中：

scp -r /usr/zookeeper root@slave1:/usr/

scp -r /usr/zookeeper root@slave2:/usr/

c.更改两个虚拟机 /usr/zookeeper/zookeeper-3.4.10/zkdata/myid 中的数字:vi /usr/zookeeper/zookeeper-3.4.10/zkdata/myid

6）配置环境变量（三台虚拟机都需要做）

vi /etc/profile

#ZOOKEEPER

export ZOOKEEPER\_HOME=/usr/zookeeper/zookeeper-3.4.10

PATH=$PATH:$ZOOKEEPER\_HOME/bin



使环境变量生效：source /etc/profile

检验：echo $PATH:

7）启动zookeeper集群（三台虚拟机都需要做）

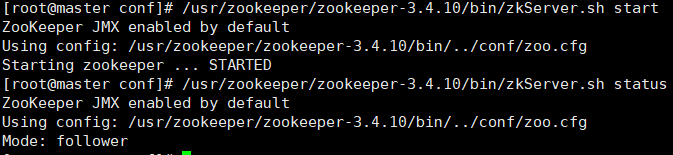
执行zookeeper的脚本命令

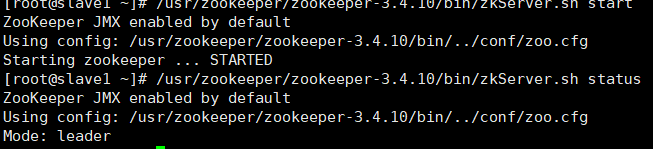
zookeeper安装目录中的bin：zkServer.sh start/status 启动/查看状态

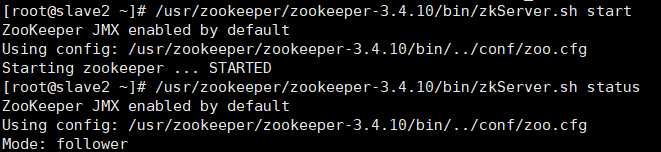
/usr/zookeeper/zookeeper-3.4.10/bin/zkServer.sh start

/usr/zookeeper/zookeeper-3.4.10/bin/zkServer.sh status

Mode: follower/leader







9、hadoop集群的搭建

1）把hadoop软件复制到虚拟机master中（先在master中操作，后面直接复制）

使用xftp工具，把 hadoop-2.7.6.tar.gz 文件拷贝到虚拟机的 /opt/soft 目录下

2）先在master这个虚拟机上安装hadoop，安装目录：/usr/hadoop

mkdir /usr/hadoop

解压文件，把/opt/soft/hadoop-2.7.6.tar.gz解压到/usr/hadoop/:

tar -xzvf /opt/soft/hadoop-2.7.6.tar.gz -C /usr/Hadoop/

3）设置环境变量

vi /etc/profile

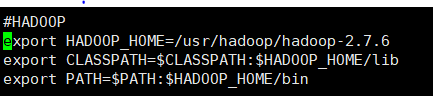
(添加以下内容)

#HADOOP

export HADOOP\_HOME=/usr/hadoop/hadoop-2.7.6

export CLASSPATH=$CLASSPATH:$HADOOP\_HOME/lib

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



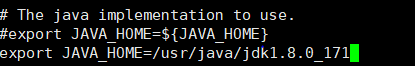
4）source /etc/profile #使得环境变量生效

5）hadoop-env.sh

vi /usr/hadoop/hadoop-2.7.6/etc/hadoop/hadoop-env.sh

把“export JAVA\_HOME=${JAVA\_HOME}”这行注释掉

并在下面添加：export JAVA\_HOME=/usr/java/jdk1.8.0\_171



6）修改core-site-xml

vi /usr/hadoop/hadoop-2.7.6/etc/hadoop/core-site.xml

在<configuration>和</configuration>之间添加

<property>

<name>fs.default.name</name>

<value>hdfs://master:9000</value>

</property>

<property>

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

<value>/usr/hadoop/hadoop-2.7.6/hdfs/tmp</value>

<description>A base for other temporary directories.</description>

</property>

<property>

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

<value>131072</value>

</property>

<property>

<name>fs.checkpoint.period</name>

<value>60</value>

</property>

<property>

<name>fs.checkpoint.size</name>

<value>67108864</value>

</property>

7）修改yarn-site-xml

vi /usr/hadoop/hadoop-2.7.6/etc/hadoop/yarn-site.xml

在<configuration>和</configuration>之间添加

<property>

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

<value>master:18040</value>

</property>

<property>

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

<value>master:18030</value>

</property>

<property>

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

<value>master:18088</value>

</property>

<property>

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

<value>master:18025</value>

</property>

<property>

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

<value>master:18141</value>

</property>

<property>

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

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>

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

</property>

8）修改hdfs-site-xml

vi /usr/hadoop/hadoop-2.7.6/etc/hadoop/hdfs-site.xml

在<configuration>和</configuration>之间添加

<property>

<name>dfs.replication</name>

<value>2</value>

</property>

<property>

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

<value>file:/usr/hadoop/hadoop-2.7.6/hdfs/name</value>

<final>true</final>

</property>

<property>

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

<value>file:/usr/hadoop/hadoop-2.7.6/hdfs/data</value>

<final>true</final>

</property>

<property>

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

<value>master:9001</value>

</property>

<property>

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

<value>true</value>

</property>

<property>

<name>dfs.permissions</name>

<value>false</value>

</property>

9）修改mapred-site-xml

复制一个样本：cp /usr/hadoop/hadoop-2.7.6/etc/hadoop/mapred-site.xml.template /usr/hadoop/hadoop-2.7.6/etc/hadoop/mapred-site.xml

vi /usr/hadoop/hadoop-2.7.6/etc/hadoop/mapred-site.xml

在<configuration>和</configuration>之间添加

<property>

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

<value>yarn</value>

</property>

10）修改/usr/hadoop/hadoop-2.7.6/etc/hadoop/slaves

vi /usr/hadoop/hadoop-2.7.6/etc/hadoop/slaves

文件内容：

删除原来的localhost

添加下面内容

slave1

slave2

11）新建/usr/hadoop/hadoop-2.7.6/etc/hadoop/master

vi /usr/hadoop/hadoop-2.7.6/etc/hadoop/master

文件内容：

master

12）slave1和slave2的安装，直接复制

scp -r /usr/hadoop root@slave1:/usr/

scp -r /usr/hadoop root@slave2:/usr/

13）格式化Hadoop(只在master上操作)

hadoop namenode -format

系统自动在hdfs/name/下创建current目录

至此，Hadoop主节点配置完成。

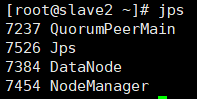
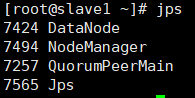
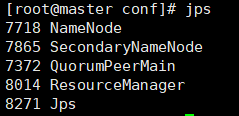
14）启动Hadoop集群（只在master上启动）

/usr/hadoop/hadoop-2.7.6/sbin/start-all.sh

注：输入jps命令查看进程状态

NameNode地址默认为50070 ，http://192.168.92.128:50070/

ResourceManager地址默认为8088



10、HBase安装(先在master上操作，之后会有复制到slave1、slave2的步骤)

前提：

a.同步时间—— systemctl status ntpd

systemctl start ntpd

ntpdate master

b.zookeeper启动(三台)——

/usr/zookeeper/zookeeper-3.4.10/bin/zkServer.sh start

/usr/zookeeper/zookeeper-3.4.10/bin/zkServer.sh status

Mode: follower/leader

c.hadoop启动（只在master）——

/usr/hadoop/hadoop-2.7.6/sbin/start-all.sh

jps命令查看hadoop启动情况（在三个虚拟机中）

1）将hbase的软件压缩包存放至/opt/soft

2）先在master这个虚拟机上安装hbase，安装目录：/usr/hbase

mkdir /usr/hbase

tar -xzvf /opt/soft/hbase-1.2.6-bin.tar.gz -C /usr/hbase/

3）在master上修改配置文件/usr/hbase/hbase-1.2.6/conf/hbase-env.sh

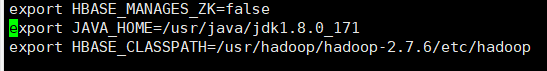
vi /usr/hbase/hbase-1.2.6/conf/hbase-env.sh

(增加内容)

export HBASE\_MANAGES\_ZK=false

export JAVA\_HOME=/usr/java/jdk1.8.0\_171

export HBASE\_CLASSPATH=/usr/hadoop/hadoop-2.7.6/etc/hadoop



4）修改配置文件 /usr/hbase/hbase-1.2.6/conf/hbase-site.xml

修改为：

<configuration>

<property>

<name>hbase.rootdir</name>

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

</property>

<property>

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

<value>true</value>

</property>

<property>

<name>hbase.master </name>

<value>hdfs://master:6000 </value>

</property>

<property>

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

<value>master,slave1,slave2 </value>

</property>

<property>

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

<value>/usr/zookeeper/zookeeper-3.4.10 </value>

</property>

</configuration>

5）修改/usr/hbase/hbase-1.2.6/conf/regionservers

删除原来的localhost

修改为：

slave1

slave2

6）把hadoop的配置文件hdfs-site.xml和core-site.xml，复制到hbase的conf目录下:

cp /usr/hadoop/hadoop-2.7.6/etc/hadoop/hdfs-site.xml /usr/hbase/hbase-1.2.6/conf/

cp /usr/hadoop/hadoop-2.7.6/etc/hadoop/core-site.xml /usr/hbase/hbase-1.2.6/conf/

7）把master上已经配置好的hbase复制到slave1和slave2

scp -r /usr/hbase root@slave1:/usr/

scp -r /usr/hbase root@slave2:/usr/

8）配置环境变量

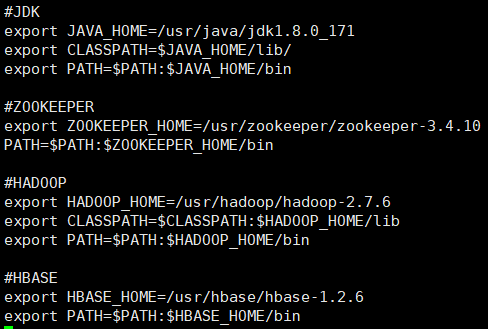
vi /etc/profile

添加如下内容:

#HBASE

export HBASE\_HOME=/usr/hbase/hbase-1.2.6

export PATH=$PATH:$HBASE\_HOME/bin

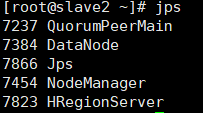
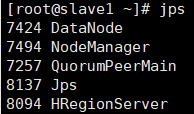
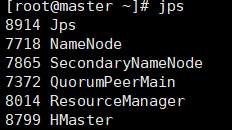


运行命令使环境变量生效：source /etc/profile

9）运行和测试

master：/usr/hbase/hbase-1.2.6/bin/start-hbase.sh

查看master节点的进程：jps HMaster/HRegionServer



11、构建数据仓库

启动：

（1）master：ntpd服务，slave1和slave2 同步时间：ntpdate master

（2）zookeeper启动：master、slave1、slave2分别启动：zkServer.sh start/status 启动/查看状态

（3）启动hadoop：master执行：/usr/hadoop/hadoop-2.7.6/sbin/start-all.sh

使用jps命令查看

（4）启动hbase：master执行：/usr/hbase/hbase-1.2.6/bin/start-hbase.sh

使用jps命令查看

slave2：安装mysql

slave1：作为hive server

master：client客户端

(1)slave2安装mysql：

a.安装mysql5.7：

安装wget：yum -y install wget

下载 mysql的repo源：wget http://repo.mysql.com/mysql57-community-release-el7-9.noarch.rpm -P /home/user/

安装mysql57-community-release-el7-9.noarch.rpm包：

rpm -ivh mysql57-community-release-el7-9.noarch.rpm

通过命令查看文件： ls -l /etc/yum.repos.d/mysql-com\*

安装mysql：yum install mysql-community-server

查看mysql的版本：mysql -V或者mysql --version

在登录mysql之后，status;



b.启动服务：

重载所有修改过的配置文件：systemctl daemon-reload

启动服务：systemctl start mysqld

或者systemctl start mysqld.service

开机自动启动mysqld：systemctl enable mysqld

c.登录mysql：

默认安装之后，mysql的root用户有密码

grep "temporary password" /var/log/mysqld.log #查看密码

mysql -h localhost -uroot -p #输入上面查看到的密码，登录mysql

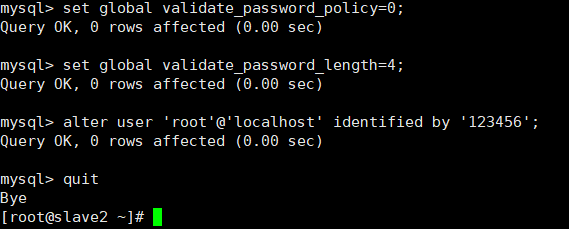
d.mysql密码安全策略设置：

设置密码强度为低级：set global validate\_password\_policy=0;

设置密码长度：set global validate\_password\_length=4;

修改本地密码： alter user 'root'@'localhost' identified by '123456';

退出mysql：quit或者\q



e.设置远程登录：

mysql -uroot -h localhost -p #登录mysql

创建远程的root用户： create user 'root'@'%' identified by '123456';

添加远程登录用户： grant all privileges on \*.\* to 'root'@'%' with grant option;

刷新权限：flush privileges;

(2)master和slave1解压hive

a.master：上传hive（apache-hive-2.1.1-bin.tar.gz）到master的/opt/soft目录中

b. mkdir /usr/hive

解压：tar -xzvf /opt/soft/apache-hive-2.1.1-bin.tar.gz -C /usr/hive/

c. 复制到slave1中： scp -r /usr/hive/ root@slave1:/usr/hive/

d. 修改master和slave1配置文件：

vi /etc/profile

#HIVE

export HIVE\_HOME=/usr/hive/apache-hive-2.1.1-bin

export PATH=$PATH:$HIVE\_HOME/bin

e. 使环境变量生效：source /etc/profile

（3）slave1作为hive server，进行配置

a、服务器需要mysql的lib安装包到Hive\_home/conf目录下

使用xftp，将驱动程序mysql-connector-java-5.1.5-bin.jar传到slave1的 /usr/hive/apache-hive-2.1.1-bin/lib

b、修改配置文件 hive-env.sh

cd /usr/hive/apache-hive-2.1.1-bin/conf/

cp hive-env.sh.template hive-env.sh

vi hive-env.sh

（修改HADOOP\_HOME的值为当前的安装位置）

HADOOP\_HOME=/usr/hadoop/hadoop-2.7.6

c、修改hive-site.xml文件

vi hive-site.xml

<configuration>

<!-- Hive产生的元数据存放位置-->

<property>

<name>hive.metastore.warehouse.dir</name>

<value>/user/hive\_remote/warehouse </value>

</property>

<!-- 数据库连接JDBC的URL地址-->

<property>

<name>javax.jdo.option.ConnectionURl</name>

<value>jdbc:mysql://slave2:3306/hive?createDatabaseIfNotExist=true</value>

</property>

<!-- 数据库连接driver,即MySQL驱动 -->

<property>

<name>javax.jdo.option.ConnectionDriverName</name>

<value>com.mysql.jdbc.Driver</value>

</property>

<!-- MySQL数据库用户名 -->

<property>

<name>javax.jdo.option.ConnectionUserName</name>

<value>root</value>

</property>

<!-- MySQL数据库密码 -->

<property>

<name>javax.jdo.option.ConnectionPassword</name>

<value>123456</value>

</property>

<property>

<name>hive.metastore.schema.verification</name>

<value>false</value>

</property>

<property>

<name>datanucleus.schema.autoCreateAll</name>

<value>true</value>

</property>

</configuration>

（4）master作为client，进行配置

a、版本冲突和jar包依赖问题

cp /usr/hive/apache-hive-2.1.1-bin/lib/jline-2.12.jar /usr/hadoop/hadoop-2.7.3/share/hadoop/yarn/lib/

b、修改hive-env.sh

cd /usr/hive/apache-hive-2.1.1-bin/conf/

cp hive-env.sh.template hive-env.sh

vi hive-env.sh

（修改HADOOP\_HOME的值为当前的安装位置）

HADOOP\_HOME=/usr/hadoop/hadoop-2.7.3

c、修改hive-site.xml文件

vi hive-site.xml

内容：

<configuration>

<!-- Hive产生的元数据存放位置-->

<property>

<name>hive.metastore.warehouse.dir</name>

<value>/user/hive\_remote/warehouse </value>

</property>

<!-- 使用本地服务连接Hive，默认为true -->

<property>

<name>hive.metastore.local</name>

<value>false</value>

</property>

<!-- 连接服务器 -->

<property>

<name>hive.metastore.uris</name>

<value>thrift://slave1:9083</value>

</property>

</configuration>

(5) 启动hive

a、slave1作为服务器，启动hive server

/usr/hive/apache-hive-2.1.1-bin/bin/hive --service metastore

b、master作为客户端，启动hive

/usr/hive/apache-hive-2.1.1-bin/bin/hive

启动成功后出现提示符：“hive>”

c、测试hive启动是否成功：

show databases;

再建立一个master的远程连接，查看进程。

jps

能够看到RunJar