操作机用户名：Administrator，密码：123456   
master用户名：root，密码：123456   
slave1用户名：root，密码：123456   
slave2用户名：root，密码：123456

**环境准备**

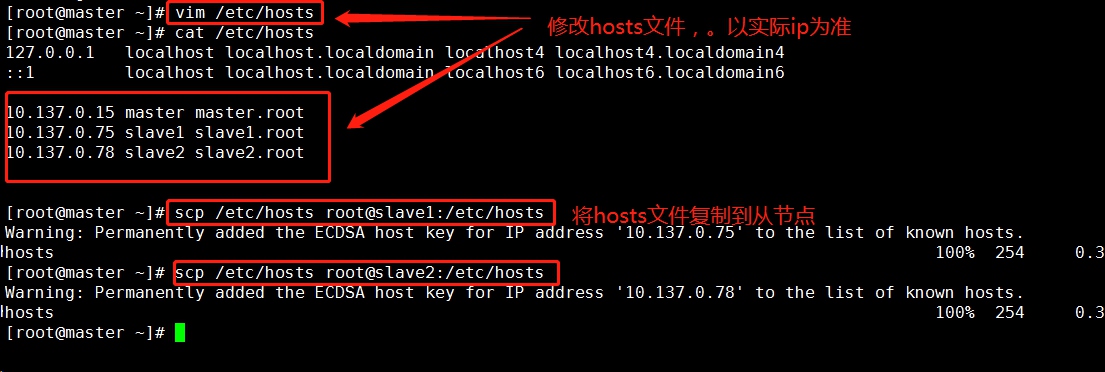
　　1.首先修改主机文件，以自己实验IP为准。（主中进行）

* vim / etc / hosts

　　然后直接将主机文件复制到从节点中。

* scp / etc / hosts root @ slave1：/ etc / hosts
* scp / etc / hosts root @ slave2：/ etc / hosts

￼



　　2.开启动物园管理员（三台机器）  
　　进入动物园管理员安装目录，直接开启动物园管理员。

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

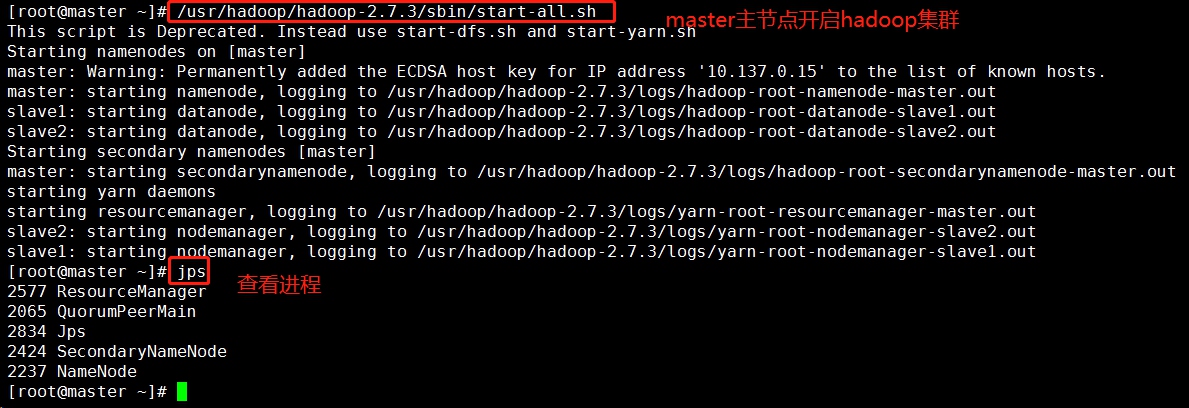
￼



　　3.开启的hadoop（主进行）

* /usr/hadoop/hadoop-2.7.3/sbin/start-all.sh

￼



　　Hadoop的集群开启之后，就可以在此基础上安装蜂箱了。

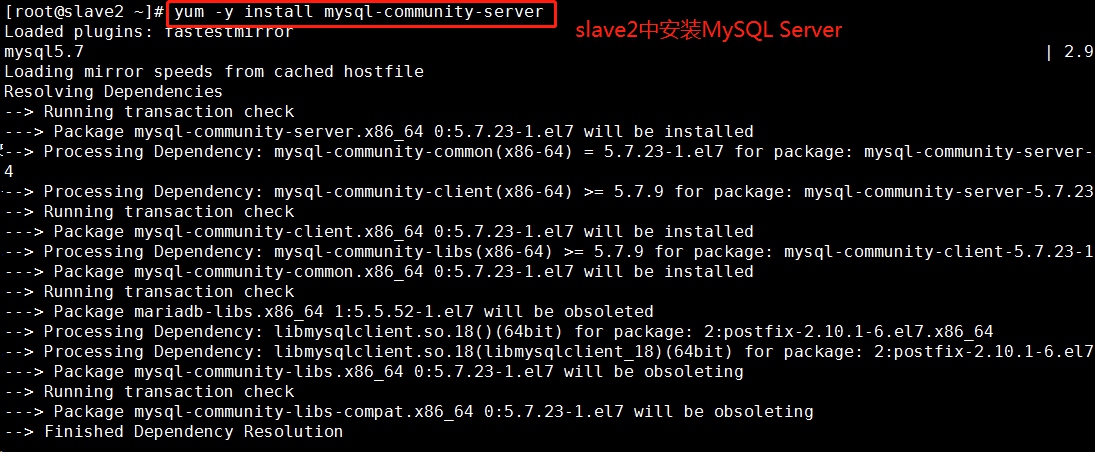
　　注意：

　　实验中我们选用hive的远程模式，slave2安装mysql server用于存放元数据，slave1作为hive server作为thrift 服务器，master作为client客户端进行操作。

**环境准备**

　　1.环境中，我们已经配置过本地源了。所以这里不用下载yum源，我们直接安装mysql server即可。

* 安装MySQL：yum -y install mysql-community-server



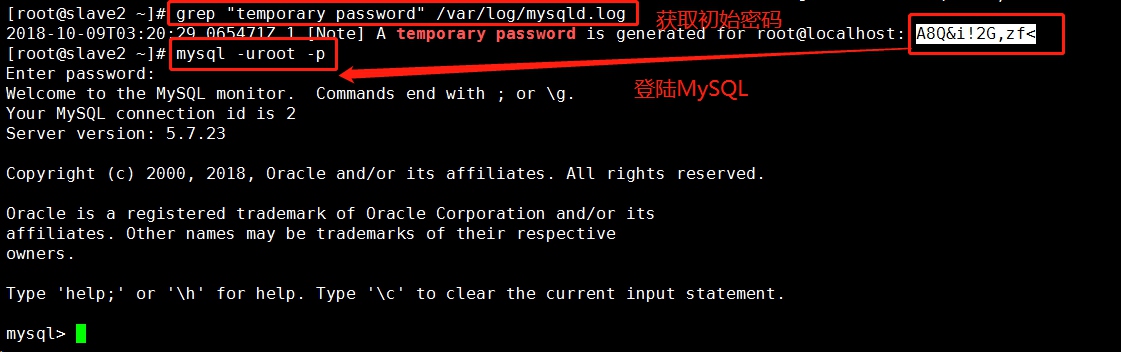
　　2.启动服务

* 重载所有修改过的配置文件：systemctl daemon-reload
* 开启服务：systemctl启动mysqld
* 开机自启：systemctl enable mysqld

http://bigdata.cloudlab.honyasec.com/storage/images/7909e155774a5ca8f900ee32b2735997/2018-10-09-11-21-25.jpg

　　3.安装完毕后，MySQL的会在/var/log/mysqld.log这个文件中会自动生成一个随机的密码，获取得这个随机密码，以用于登录的MySQL数据库：

* 获取初密码：grep“临时密码”/var/log/mysqld.log
* 登陆MySQL：mysql -uroot -p（注意中英文）



　　4.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：由'123456'标识的用户'root'@'localhost';
* 退出：\ q

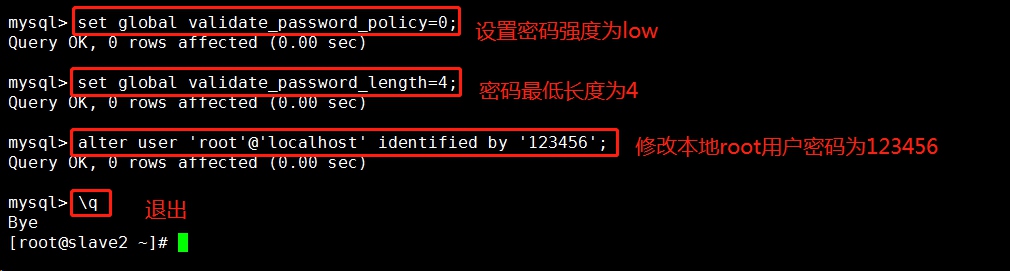
密码强度分级如下：

0为low级别，只检查长度；

1为medium级别（默认），符合长度为8，且必须含有数字，大小写，特殊字符;

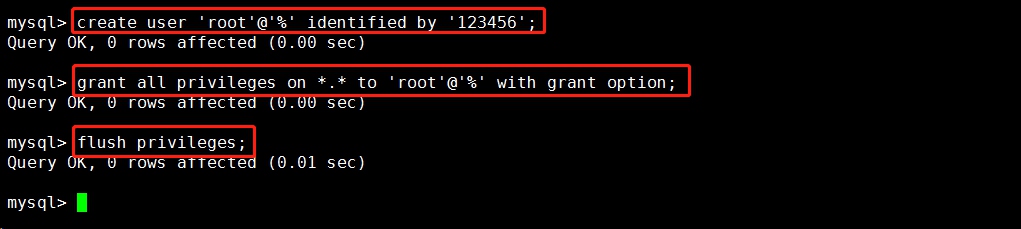
2为strong级别，密码难度更大一些，需要包括字典文件。

密码长度最低长为4，当设置长度为1、2、3时，其长度依然为4。



　　5.设置远程登录

* 以新密码登陆MySQL：mysql -uroot -p123456
* 创建root用户：创建'123456'标识的用户'root'@'％';
* 添加远程登录用户：使用grant选项将\*。\*的所有权限授予'root'@'％';
* 刷新权限：刷新权限;



上一步

**环境准备**

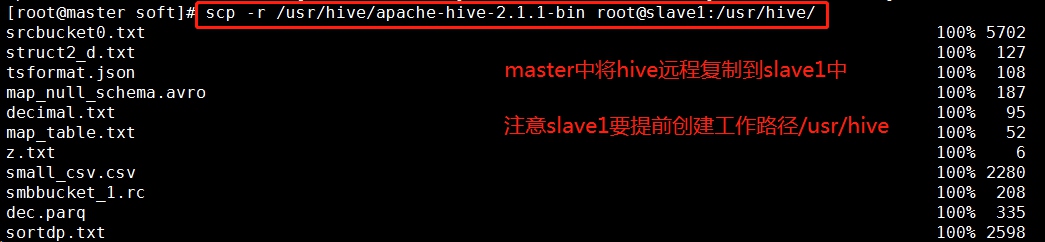
　　1.首先我们需要创建工作路径，并将hive解压。环境中master作为客户端，slave1作为服务器端，因此都需要使用到hive。因为hive相关安装包存放在master中，因此我们先在master中对hive进行解压，然后将其复制到slave1中。  
　　master中操作如下：

* cd /opt/soft
* ls
* mkdir -p /usr/hive
* tar -zxvf /opt/soft/apache-hive-2.1.1-bin.tar.gz -C /usr/hive/



　　2.同样slave1上建立文件夹/usr/hive，master中将安装包远程复制到slave1。

* scp -r /usr/hive/apache-hive-2.1.1-bin root@slave1:/usr/hive/



3.修改/etc/profile文件设置hive环境变量。（master和slave1都执行）

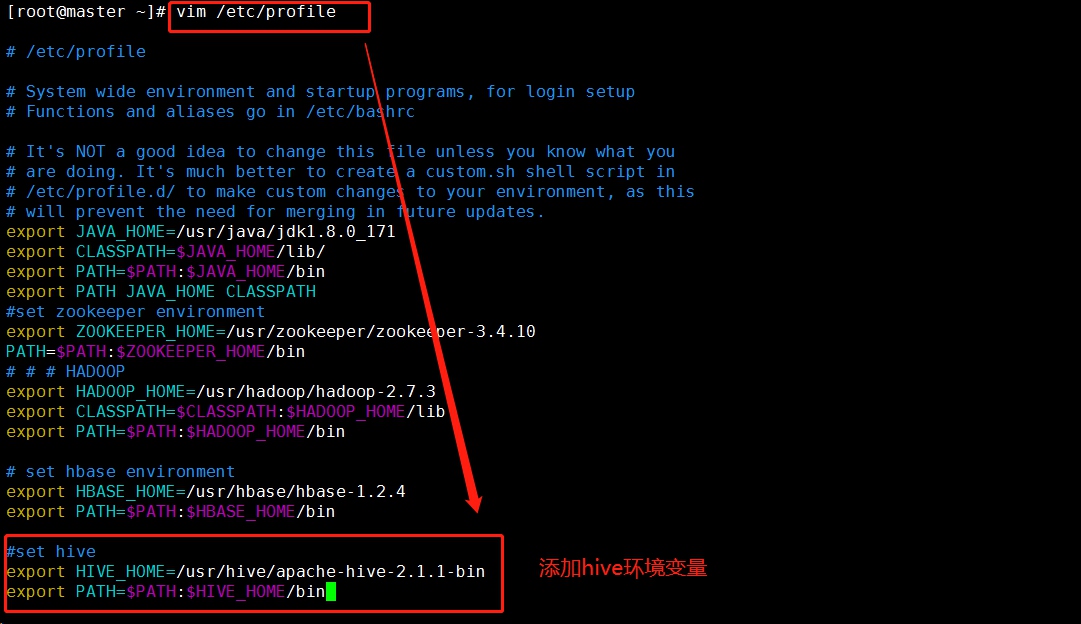
**master**

* vim /etc/profile

#set hive

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

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



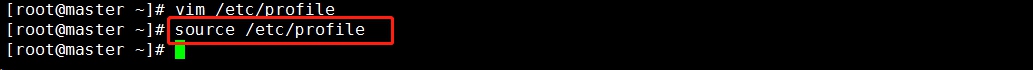
生效环境变量：

**salve1**

export HIVE\_HOME=/usr/hive/

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

* source /etc/profile

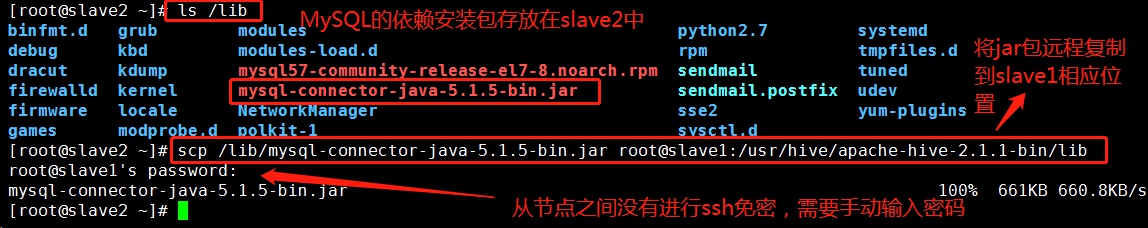


上一步

**slave1作为hive server，进行配置**

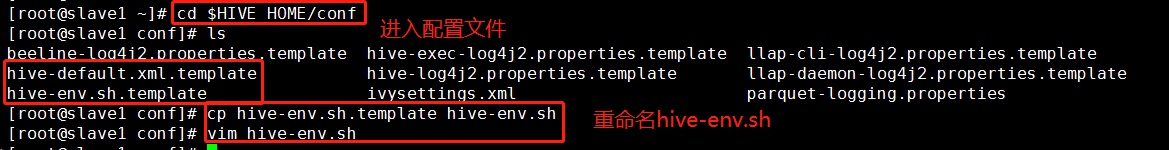
　　1.因为服务器端需要和Mysql通信，所以服务器端需要Mysql的lib安装包到Hive\_Home/conf目录下。  
　　注意：mysql.jar放在slave2中的/lib目录下，需要将其远程复制到slave1的hive的lib中。  
　　首先slave2中进行如下操作：

* 查看lib：ls /lib
* 远程复制依赖包到slave1：~~scp /lib/mysql-connector-java-5.1.5-bin.jar~~ [~~root@slave1:/usr/hive/apache-hive-2.1.1-bin/lib~~](mailto:root@slave1:/usr/hive/apache-hive-2.1.1-bin/lib)
* scp /lib/mysql-connector-java-5.1.5-bin.jar root@slave1:/usr/hive/lib



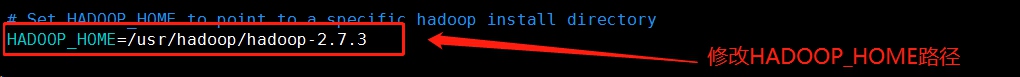
　　2.回到slave1，修改hive-env.sh文件中HADOOP\_HOME环境变量。进入hive配置目录，因为hive中已经给出配置文件的范本hive-env.sh.template，直接将其复制一个进行修改即可：

* cd $HIVE\_HOME/conf
* ls
* cp hive-env.sh.template hive-env.sh
* vim hive-env.sh



　　hive-env.sh文件中修改HADOOP\_HOME环境变量，内容如下：

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



　　3.修改hive-site.xml文件

* vim hive-site.xml

　　添加内容如下：

<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> #连接MySQL所在的ip（主机名）及端口

</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>

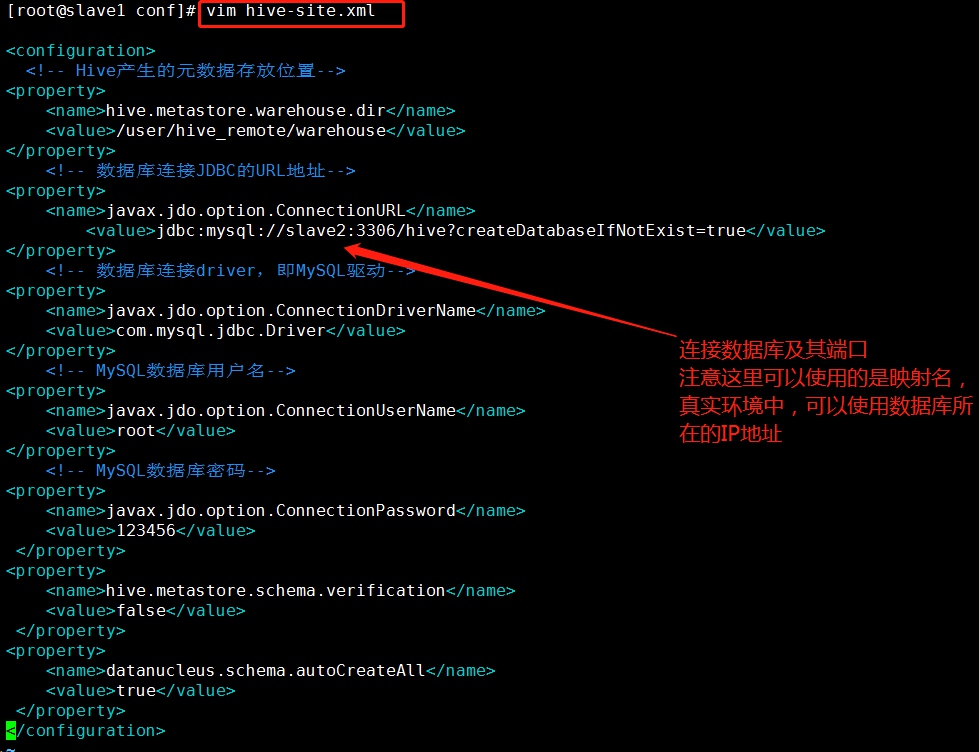
<property>

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

<value>true</value>

</property>

</configuration>



上一

**master作为客户端，进行配置**

　　1.解决版本冲突和jar包依赖问题。  
　　由于客户端需要和Hadoop通信，所以需要更改Hadoop中jline的版本。即保留一个高版本的jline jar包，从hive的lib包中拷贝到Hadoop中lib位置为/usr/hadoop/hadoop-2.7.3/share/hadoop/yarn/lib。

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

http://bigdata.cloudlab.honyasec.com/storage/images/7909e155774a5ca8f900ee32b2735997/2018-10-09-13-38-21.jpg

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

2.修改hive-env.sh中HADOOP\_HOME环境变量：

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

　　3.修改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>

<!--- 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-->

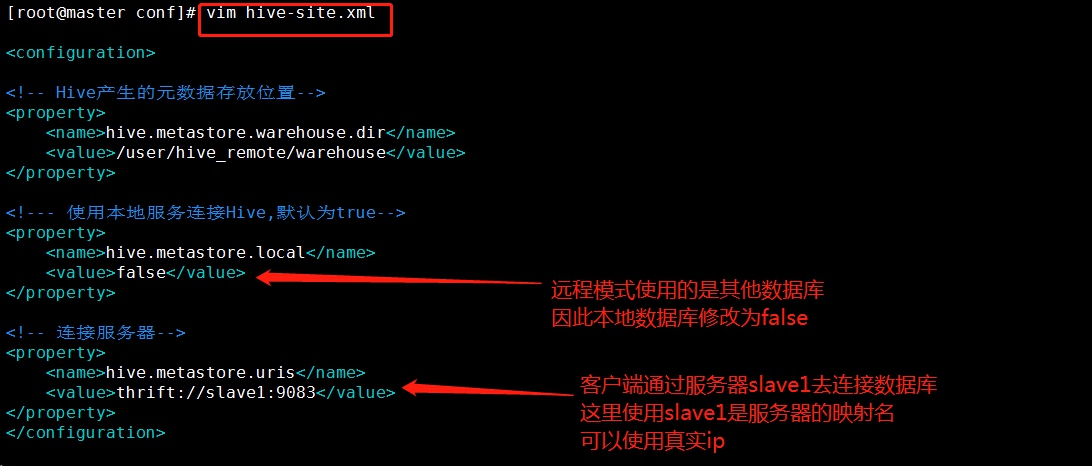
<property>

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

<value>thrift://slave1:9083</value> #hive客户端通过thrift服务器服务连接MySQL数据库，这里的thrift服务器就是slave1的ip（主机名）

</property>

</configuration>

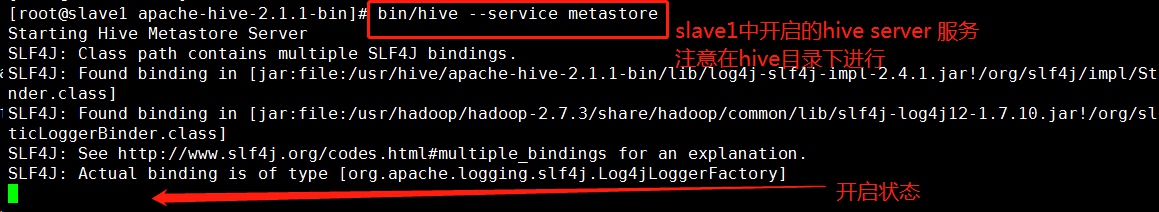


上一步

**启动hive**

　　1.slave1作为服务器端，开启hive server。

* bin/hive --service metastore

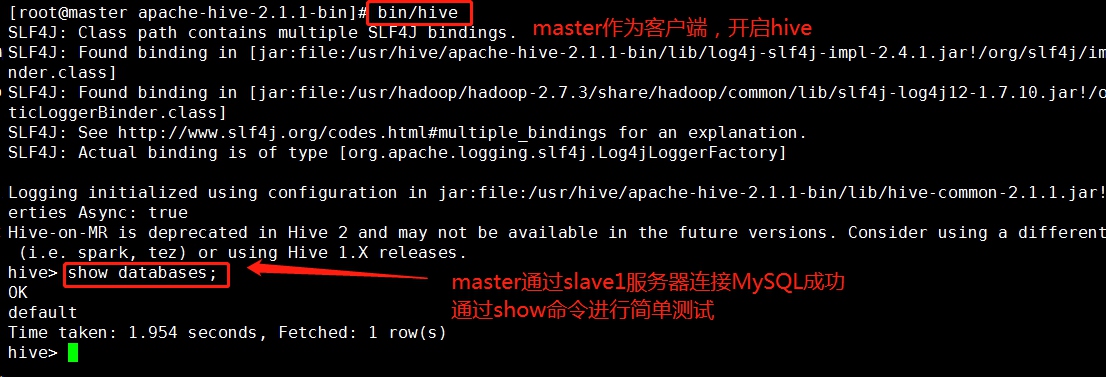


　　2.master作为客户端，启动hive。

* bin/hive

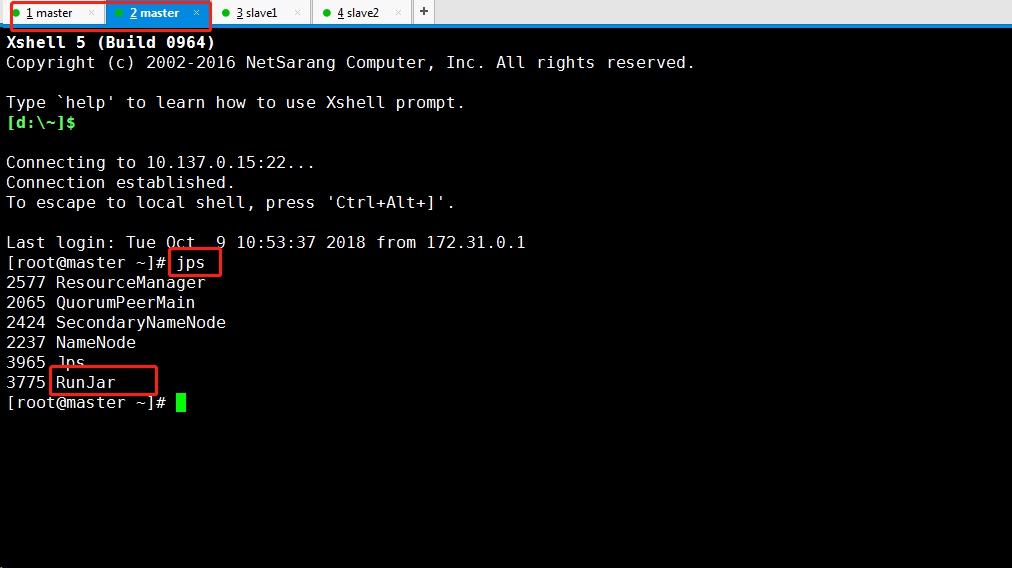
　　测试hive是否启动成功：

* hive>show databases;



　　3.最后复制一个master，查看其进程如下：

* jps

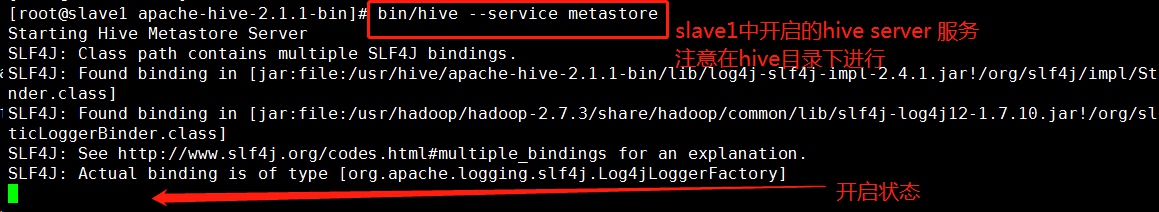


上

实验总结空空的

　　1.slave1作为服务器端，开启hive server。

* bin/hive --service metastore

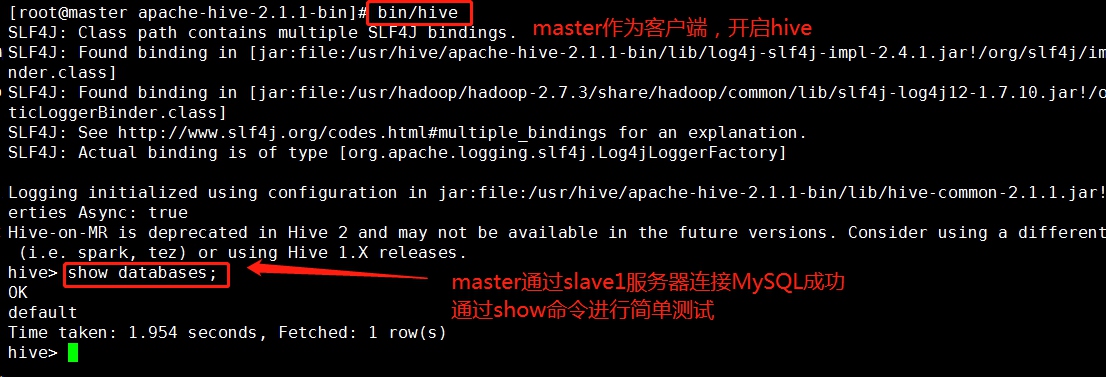


　　2.master作为客户端，启动hive。

* bin/hive

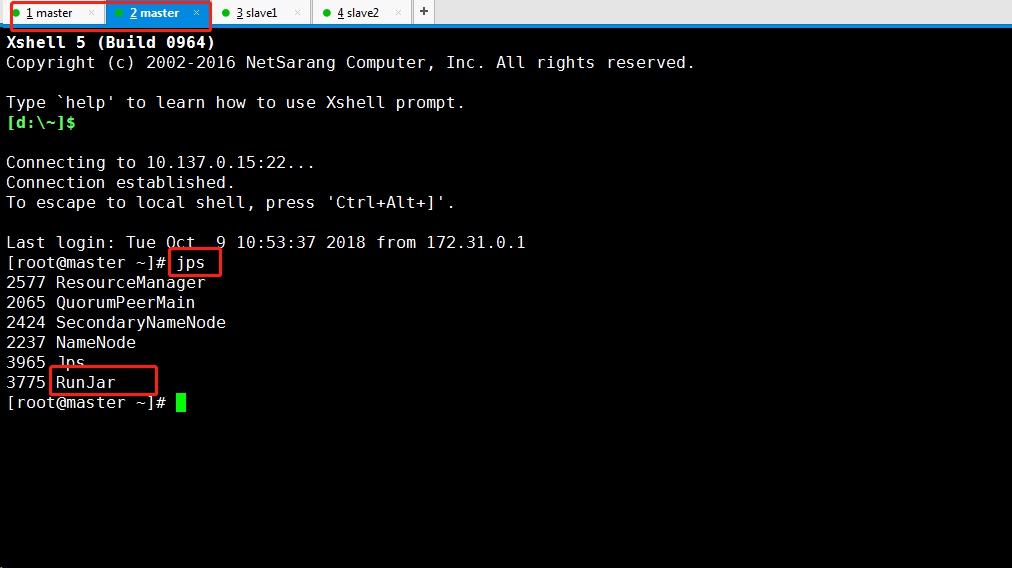
　　测试hive是否启动成功：

* hive>show databases;



　　3.最后复制一个master，查看其进程如下：

* jps



上一步