Hive的三种安装方式（内嵌模式，本地模式远程模式）

**一、安装模式介绍：**

Hive官网上介绍了Hive的3种安装方式，分别对应不同的应用场景。

1、内嵌模式（元数据保村在内嵌的derby种，允许一个会话链接，尝试多个会话链接时会报错）

2、本地模式（本地安装mysql 替代derby存储元数据）

3、远程模式（远程安装mysql 替代derby存储元数据）

**二、安装环境以及前提说明：**

首先，Hive是依赖于hadoop系统的，因此在运行Hive之前需要保证已经搭建好hadoop集群环境。

本文中使用的hadoop版本为2.6.5，Hive版本为2.1.1版。

OS：Linux Centos 6.7 64位

jdk：java version "1.8.0\_151"

假设已经下载了Hive的安装包，且安装到了/opt/bigdata/hive  
在/etc/profile中设定HIVE\_HOME环境变量：

[hadoop@master hive]$ sudo vi /etc/profile

添加：

export HIVE\_HOME=/opt/bigdata/hive

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

立即生效：

[hadoop@master hive]$ source /etc/profile

**三、内嵌模式安装：**

这种安装模式的元数据是内嵌在Derby数据库中的，只能允许一个会话连接，数据会存放到HDFS上。

1、切换到HIVE\_HOME/conf目录下，执行下面的命令：

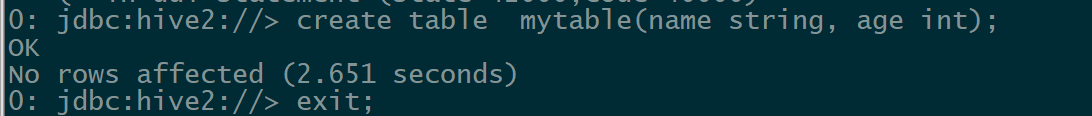
cp hive-env.sh.template hive-env.sh

vim hive-env.sh

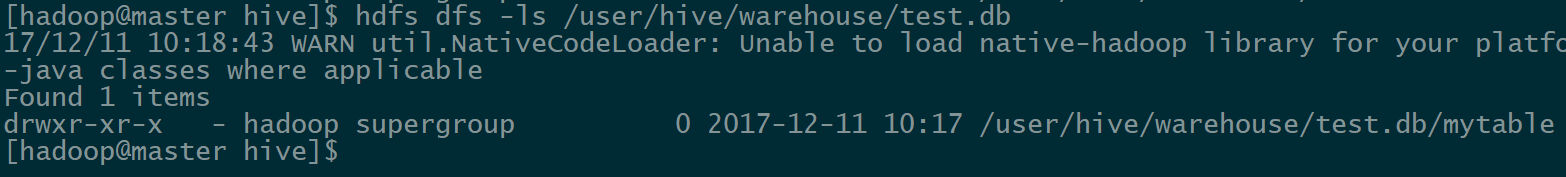
在hive-env.sh中添加以下内容：

HADOOP\_HOME=/opt/bigdata/hadoop

2、启动hive，由于已经将HIVE\_HOME加入到了环境变量中，所以这里直接在命令行敲hive即可：



然后我们看到在hadoop的HDFS上已经创建了对应的目录。



注意，只要上面2步即可完成内嵌模式的安装和启动

**四、本地模式安装：**

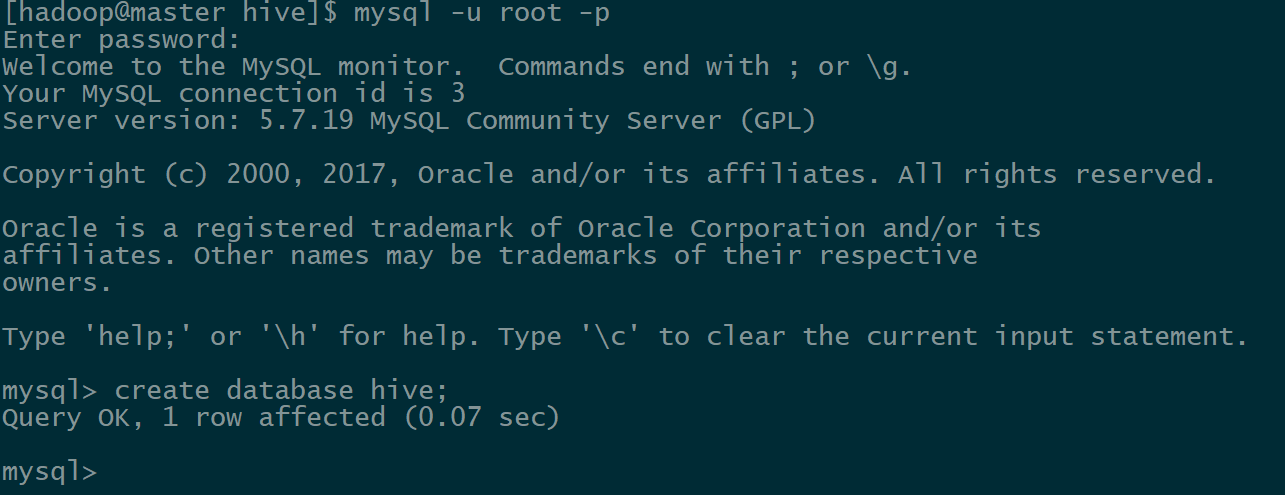
这种安装方式和嵌入式的区别在于，不再使用内嵌的Derby作为元数据的存储介质，而是使用其他数据库比如MySQL来存储元数据。

这种方式是一个多用户的模式，运行多个用户client连接到一个数据库中。这种方式一般作为公司内部同时使用Hive。

这里有一个前提，每一个用户必须要有对MySQL的访问权利，即每一个客户端使用者需要知道MySQL的用户名和密码才行。

下面开始正式搭建，这里要求hadoop系统已经正常启动，且MySQL数据库已经正确安装。

1、首先登录MySQL，创建一个数据库，这里命名为hive，数据库名是可以随意定义的。



创建hive用户，并赋予所有的权限：

CREATE USER 'hive'@'master' IDENTIFIED BY '123456';

GRANT ALL PRIVILEGES ON \*.\* TO hive IDENTIFIED BY '123456'  WITH GRANT OPTION;

2、将MySQL的JDBC驱动包拷贝到hive的安装目录中，驱动包自行查找下载。

[hadoop@master pkg]$ cp ./mysql-connector-java-5.1.31-bin.jar /opt/bigdata/hive/lib/

3、将HIVE\_HOME/conf下的hive-default.xml.template拷贝一份：

[hadoop@master pkg]$ cd /opt/bigdata/hive/conf/

[hadoop@master conf]$ cp hive-default.xml.template hive-site.xml

4、修改hive-site.xml文件：

该配置文件有3300多行，选择其中的几个选项进行修改即可。

A、修改javax.jdo.option.ConnectionURL属性。

<property>

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

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

<description>JDBC connect string for a JDBC metastore</description>

</property>

B、修改javax.jdo.option.ConnectionDriverName属性。

<property>

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

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

<description>Driver class name for a JDBC metastore</description>

</property>

C、修改javax.jdo.option.ConnectionUserName属性。即数据库用户名。

<property>

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

<value>hive</value>

<description>Username to use against metastore database</description>

</property>

D、修改javax.jdo.option.ConnectionPassword属性。即数据库密码。

<property>

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

<value>123456</value>

<description>password to use against metastore database</description>

</property>

E、添加如下属性hive.metastore.local：

<property>

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

<value>true</value>

<description>controls whether to connect to remove metastore server or open a new metastore server in Hive Client JVM</description>

</property>

F、修改hive.server2.logging.operation.log.location属性，因为默认的配置里没有指定具体的路径。

<property>

<name>hive.server2.logging.operation.log.location</name>

<value>/tmp/logs/hive</value>

<description>

Top level directory where operation logs are stored if logging functionality is enabled

</description>

</property>

G、修改hive.exec.local.scratchdir属性。

<property>

<name>hive.exec.local.scratchdir</name>

<value>/tmp/hive/scratch</value>

<description>Local scratch space for Hive jobs</description>

</property>

H、修改hive.downloaded.resources.dir属性。

<property>

<name>hive.downloaded.resources.dir</name>

<value>/tmp/hive/resources</value>

<description>Temporary local directory for added resources in the remote file system.</description>

</property>

I、修改属性hive.querylog.location属性。

<property>

<name>hive.querylog.location</name>

<value>/tmp/hive/querylog</value>

<description>Location of Hive run time structured log file</description>

</property>

5、创建目录。

hdfs dfs -mkdir -p /tmp/hive/scratch

hdfs dfs -mkdir -p /tmp/hive/resources

hdfs dfs -mkdir -p /tmp/hive/querylog

hdfs dfs -chmod -R 777 /tmp/hive/scratch

hdfs dfs -chmod -R 777 /tmp/hive/resources

hdfs dfs -chmod -R 777 /tmp/hive/querylog

6、配置hive的log4j配置文件。

[hadoop@master conf]$ cp hive-log4j2.properties.template hive-log4j.properties

[hadoop@master conf]$ vi hive-log4j.properties

property.hive.log.dir = ${sys:java.io.tmpdir}/${sys:user.name}

改成：

property.hive.log.dir = /tmp/logs/hive/

[hadoop@master conf]$ cp beeline-log4j2.properties.template beeline-log4j2.properties

7、将hive下的jline-2.12.jar替换掉hadoop自带的包，不然会报错。

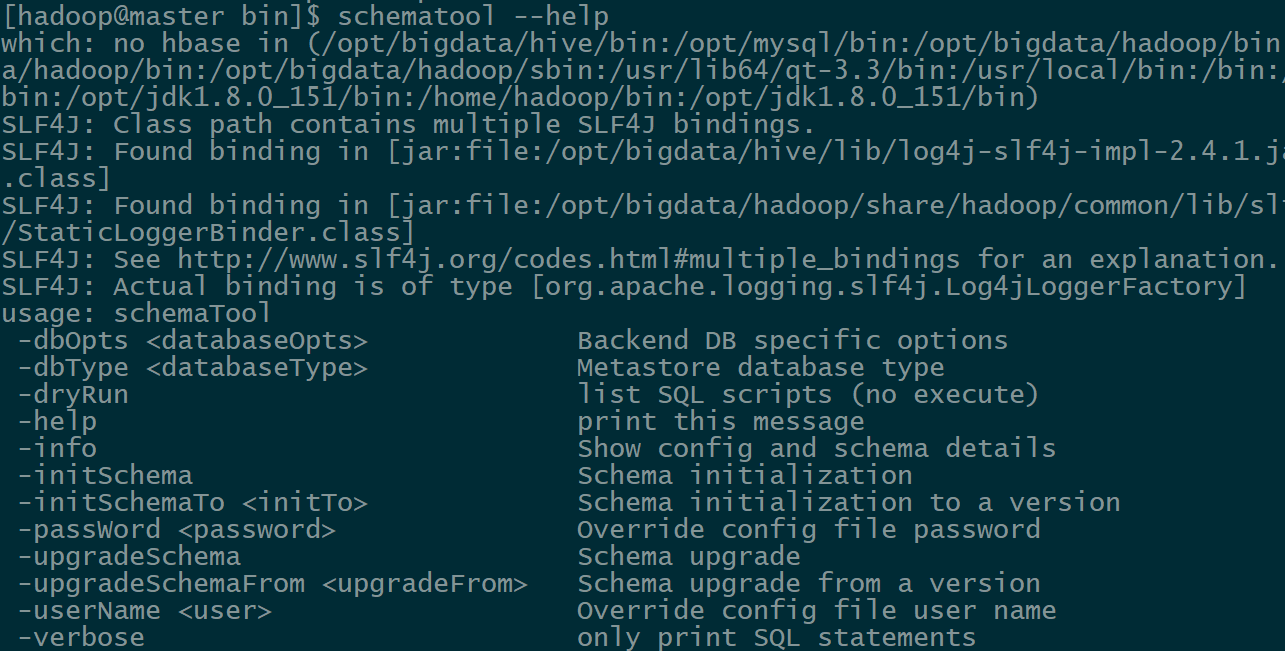
[hadoop@master lib]$ cp /opt/bigdata/hive/lib/jline-2.12.jar /opt/bigdata/hadoop/share/hadoop/yarn/lib/

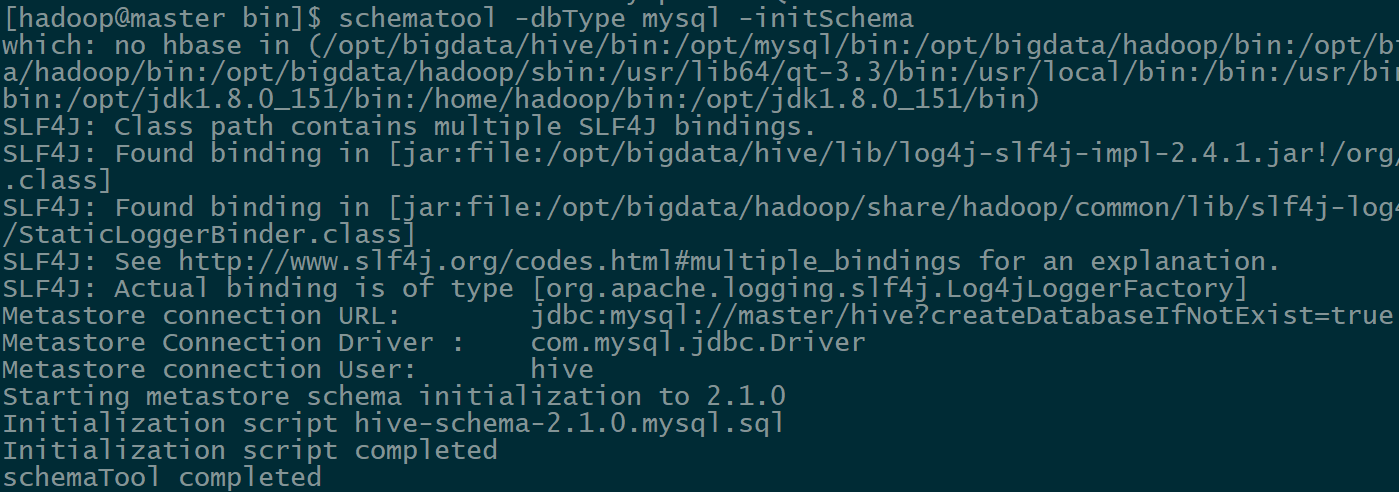
[hadoop@master lib]$ rm /opt/bigdata/hadoop/share/hadoop/yarn/lib/jline-0.9.94.jar

8、hive2元数据库初始化

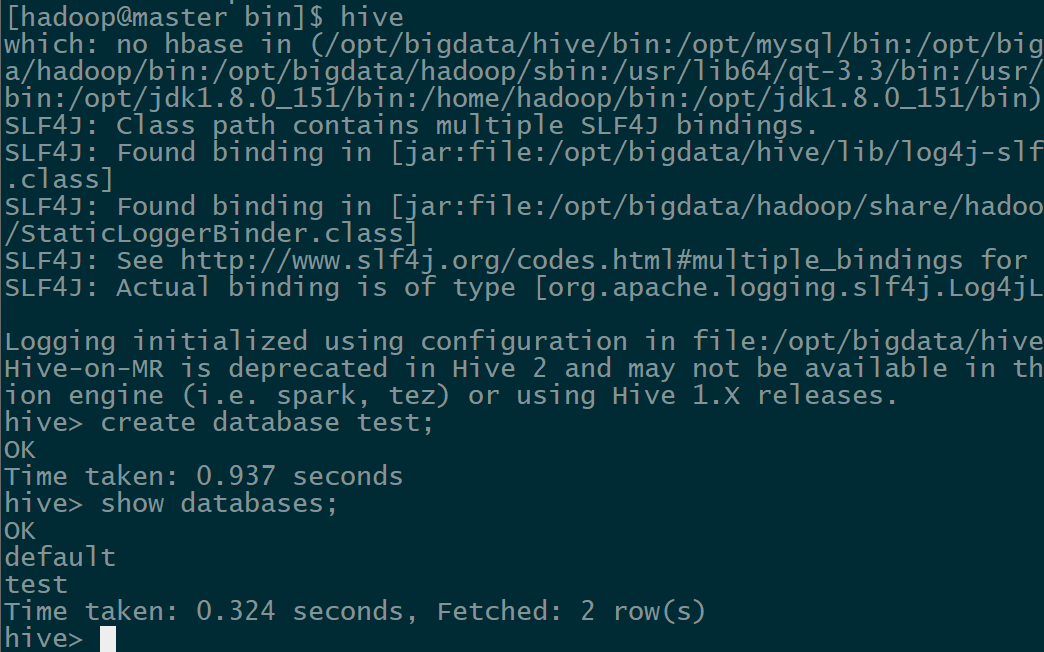
[hadoop@master bin]$ schematool –help

[hadoop@master bin]$ schematool -dbType mysql -initSchema





9、启动hive，并测试：



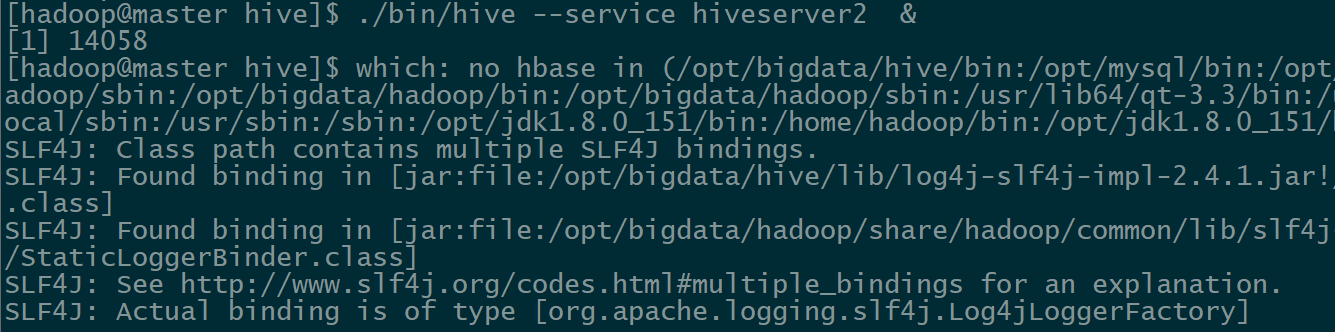
**五、远程模式安装，即server模式。**

这种模式需要使用hive安装目录下提供的beeline+hiveserver2配合使用才可以。

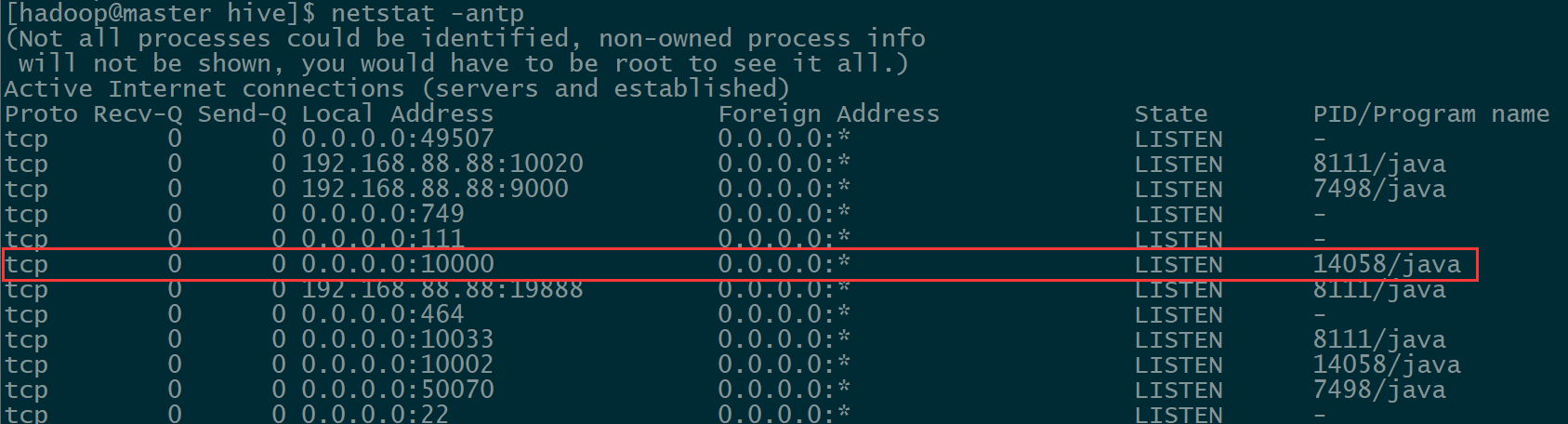
其原理就是将metadata作为一个单独的服务进行启动。各种客户端通过beeline来连接，连接之前无需知道数据库的密码。

1、首先执行hiveserver2命令：

[hadoop@master hive]$ ./bin/hive --service hiveserver2 &



启动后命令行会一直监听不退出，我们可以看到它监听了10000端口。



2、新开一个命令行窗口，执行beeline命令：

[hadoop@master hive]$ bin/beeline

beeline> !connect jdbc:hive2://

Connecting to jdbc:hive2://

Enter username for jdbc:hive2://: hive

Enter password for jdbc:hive2://: \*\*\*\*\*\*

连接：成功了。

