[Hive的三种安装方式（内嵌模式，本地模式远程模式）](http://www.cnblogs.com/tq03/p/5107949.html)

**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2.5.1，Hive版本为1.2.1版。

OS：Linux Centos 6.5 64位

jdk：java version "1.7.0\_79"

假设已经下载了Hive的安装包，且安装到了/home/install/hive-1.2.1   
在~/.bash\_profile中设定HIVE\_HOME环境变量：

export HIVE\_HOME=/home/install/hive-1.2.1

**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：

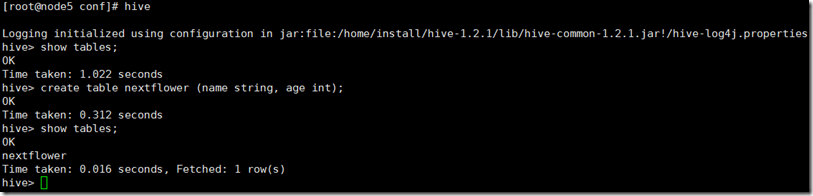
cp hive-env.sh.template hive-env.sh

vim hive-env.sh

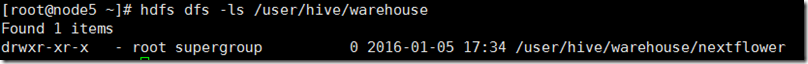
在hive-env.sh中添加以下内容：

HADOOP\_HOME=/home/install/hadoop-2.5.1

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：

[](http://images2015.cnblogs.com/blog/587689/201601/587689-20160107002419481-1258143201.png)

然后我们看到在hadoop的HDFS上已经创建了对应的目录。

[](http://images2015.cnblogs.com/blog/587689/201601/587689-20160107002420371-517916671.png)

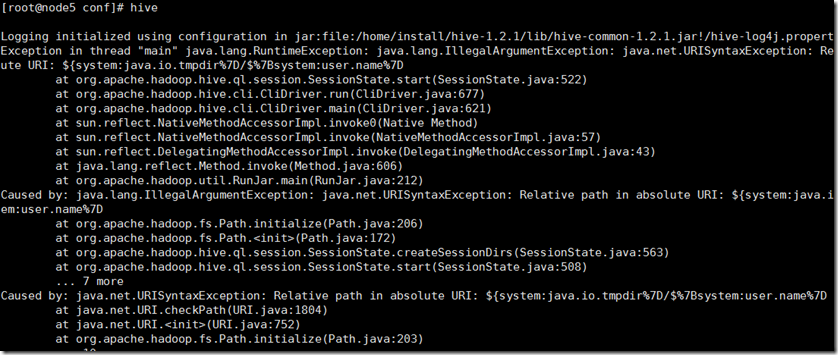
注意，只要上面2步即可完成内嵌模式的安装和启动，不要画蛇添足。。。。。。比如下面

============================【**下面这段就不要看了**】========================================

（**作废**）2、提供一个hive的基础配置文件，执行如下代码，就是将conf目录下自带的文件修改为配置文件：

cp hive-default.xml.template hive-site.xml

（**作废**）3、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：

[](http://images2015.cnblogs.com/blog/587689/201601/587689-20160107002421700-151851037.png)

（**作废**）结果报错了，看错误日志中提到system:java.io.tmpdir，这个配置项在hive-site.xml中有提到。

（**作废**）我们创建一个临时目录/opt/tem/hive-1.2.1/iotemp，并修改hive-site.xml中的system:java.io.tmpdir的值：

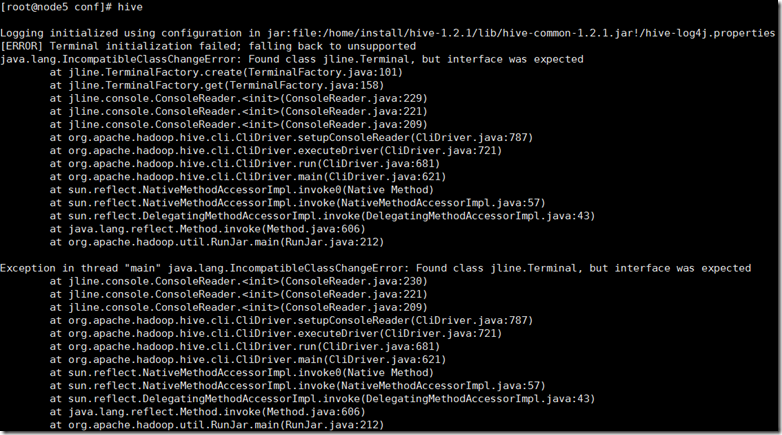
mkdir -p /opt/tem/hive-1.2.1/iotemp

vim hive-site.xml

（**作废**）在vim编辑界面输入如下命令完成替换：

:%s@\${system:java.io.tmpdir}@/opt/tem/hive-1.2.1/iotemp@g

（**作废**）4、重新启动hive：

[](http://images2015.cnblogs.com/blog/587689/201601/587689-20160107002423965-1310982739.png)

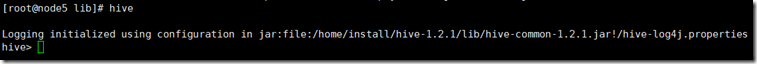
（**作废**）报了这样一个错误：java.lang.IncompatibleClassChangeError: Found class jline.Terminal, but interface was expected。

（**作废**）查询资料说，hadoop目录下存在老版本jline，替换掉就行了。**复制后注意删除原来版本的jar包**。

cp /home/install/hive-1.2.1/lib/jline-2.12.jar /home/install/hadoop-2.5.1/share/hadoop/yarn/lib/

   rm -rf /home/install/hadoop-2.5.1/share/hadoop/yarn/lib/jline-0.9.94.jar

（**作废**）再次重新启动，OK了。

[](http://images2015.cnblogs.com/blog/587689/201601/587689-20160107002426496-598046122.png)

**四、本地模式安装：**

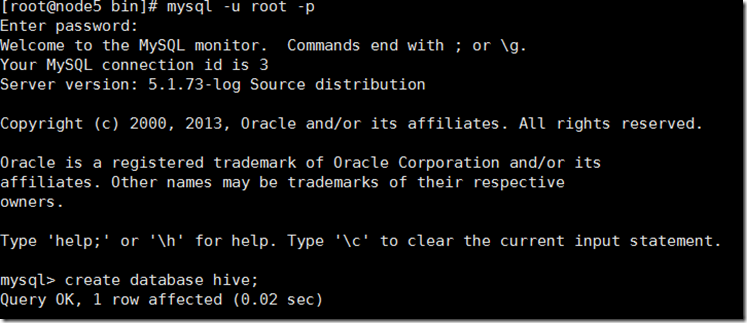
这种安装方式和嵌入式的区别在于，不再使用内嵌的Derby作为元数据的存储介质，而是使用其他数据库比如MySQL来存储元数据。

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。

[](http://images2015.cnblogs.com/blog/587689/201601/587689-20160107002427762-1604740414.png)

创建hive用户，并赋予所有的权限：

CREATE USER 'hive'@'localhost' 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。

cp mysql-connector-java-5.1.32-bin.jar /home/install/hive-1.2.1/lib/

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

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://localhost/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/hive/operation\_logs</value>

<description>Top level directory where operation logs are stored if logging functionality is enabled</descripti on>

</property>

G、修改hive.exec.local.scratchdir属性。

<property>

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

<value>/tmp/hive</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、配置hive的log4j配置文件。

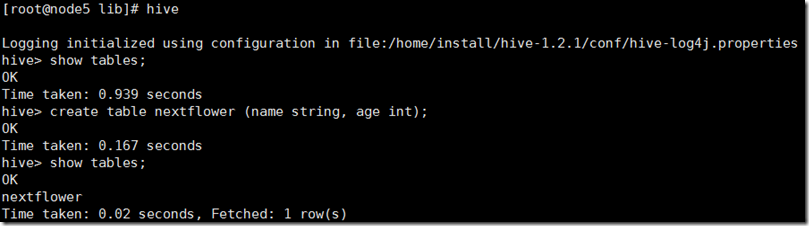
cp hive-log4j.properties.template hive-log4j.properties

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

cp /home/install/hive-1.2.1/lib/jline-2.12.jar /home/install/hadoop-2.5.1/share/hadoop/yarn/lib/

rm -rf /home/install/hadoop-2.5.1/share/hadoop/yarn/lib/jline-0.9.94.jar

7、启动hive，界面如下：

[](http://images2015.cnblogs.com/blog/587689/201601/587689-20160107002429262-1350813251.png)

**五、远程模式安装，即server模式。**

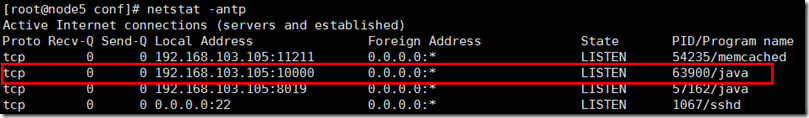
这种模式需要使用hive安装目录下提供的beeline+hiveserver2配合使用才可以。

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：

./hiveserver2 start

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。

[](http://images2015.cnblogs.com/blog/587689/201601/587689-20160107002430215-1941329751.png)

2、新开一个命令行窗口，执行beeline命令：

[root@node5 bin]# beeline

Beeline version 1.2.1 by Apache Hive

beeline> !connect jdbc:hive2://node5:10000

Connecting to jdbc:hive2://node5:10000

Enter username for jdbc:hive2://node5:10000: hive

Enter password for jdbc:hive2://node5:10000: \*\*\*\*\*\*

报错了，错误日志如下：

[复制代码](javascript:void(0);)

Error: Failed to open new session: java.lang.RuntimeException: java.lang.RuntimeException: org.apache.hadoop.security.AccessControlException: Permission denied: user=hive, access=EXECUTE, inode="/tmp":root:supergroup:drwx------

at org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.checkFsPermission(FSPermissionChecker.java:271)

at org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.check(FSPermissionChecker.java:257)

at org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.checkTraverse(FSPermissionChecker.java:208)

at org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.checkPermission(FSPermissionChecker.java:171)

at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.checkPermission(FSNamesystem.java:5904)

at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.getFileInfo(FSNamesystem.java:3691)

at org.apache.hadoop.hdfs.server.namenode.NameNodeRpcServer.getFileInfo(NameNodeRpcServer.java:803)

at org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolServerSideTranslatorPB.getFileInfo(ClientNamenodeProtocolServerSideTranslatorPB.java:779)

at org.apache.hadoop.hdfs.protocol.proto.ClientNamenodeProtocolProtos$ClientNamenodeProtocol$2.callBlockingMethod(ClientNamenodeProtocolProtos.java)

at org.apache.hadoop.ipc.ProtobufRpcEngine$Server$ProtoBufRpcInvoker.call(ProtobufRpcEngine.java:585)

at org.apache.hadoop.ipc.RPC$Server.call(RPC.java:928)

at org.apache.hadoop.ipc.Server$Handler$1.run(Server.java:2013)

at org.apache.hadoop.ipc.Server$Handler$1.run(Server.java:2009)

at java.security.AccessController.doPrivileged(Native Method)

at javax.security.auth.Subject.doAs(Subject.java:415)

at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1614)

at org.apache.hadoop.ipc.Server$Handler.run(Server.java:2007) (state=,code=0)

0: jdbc:hive2://node5:10000 (closed)>

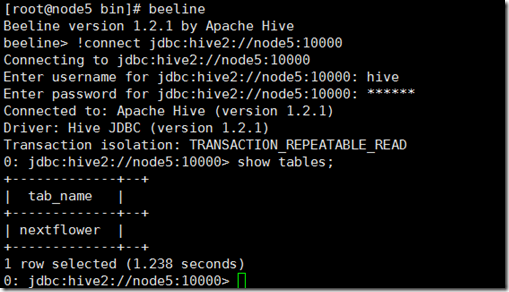
[复制代码](javascript:void(0);)

说是hive用户对HDFS上的/tmp权限不够。

这里直接把HDFS的权限设置为最大。

hadoop fs -chmod 777 /tmp

重新连接：成功了。

[](http://images2015.cnblogs.com/blog/587689/201601/587689-20160107002431090-2065959408.png)