在线实验，请到PC端体验

Hive安装配置 (没看)

一、实验介绍

1.1 实验内容

- Hive 安装与准备
- Hive 的运行模式
- Hive 与mysql数据库的连接

1.2 实验知识点

- 配置数据库连接驱动
- 运行模式
- mysql

1.3 实验环境

- Hive V2.0.0
- hadoop2.4.1
- Xfce终端

1.4 适合人群

本课程难度为一般，属于初级级别课程，适合具有hadoop基础的用户，熟悉linux基础知识

二、Hive 运行模式

与 Hadoop 类似，Hive 也有 3 种运行模式：

> **1. 内嵌模式**
>
> 将元数据保存在本地内嵌的 Derby 数据库中，这是使用 Hive 最简单的方式。但是这种方式缺点也比较明显，因为一个内嵌的 Derby 数据库每次只能访问一个数据文件，这也就意味着它不支持多会话连接。
>
> **2. 本地模式**
>
> 这种模式是将元数据保存在本地独立的数据库中（一般是 MySQL），这用就可以支持多会话和多用户连接了。
>
> **3. 远程模式**
>
> 此模式应用于 Hive 客户端较多的情况。把 MySQL 数据库独立出来，将元数据保存在远端独立的 MySQL 服务中，避免了在每个客户端都安装 MySQL 服务从而造成冗余浪费的情况。

三、下载安装 Hive（该步骤已由实验楼完成）

上节课程我们已经了解到，Hive 是基于 Hadoop 文件系统之上的数据仓库。因此，安装Hive之前必须确保 Hadoop 已经成功安装。本次实验，使用Hive V2.0.0 版本。Hive V2.0.0 可以在 Hadoop V2.4.x 以上环境中工作。

- 下载apache-hive-2.0.0-bin.tar.gz 后，对其进行解压：

  ```
  tar zxvf apache-hive-2.0.0-bin.tar.gz
  ```

  也可以在/opt目录中找到解压好的包。

四、配置系统环境变量（该步骤已由实验楼完成）

修改/etc/profile文件，这个我们在 Hadoop 和 HBase 的课程 (http://www.shiyanlou.com/courses/37) 中也修改过，应该比较熟悉了。

```
sudo vim /home/hadoop/.bashrc
# Hive environment
    export HIVE_HOME=/opt/apache-hive-2.0.0-bin
    export PATH=$PATH:$HIVE_HOME/bin
```

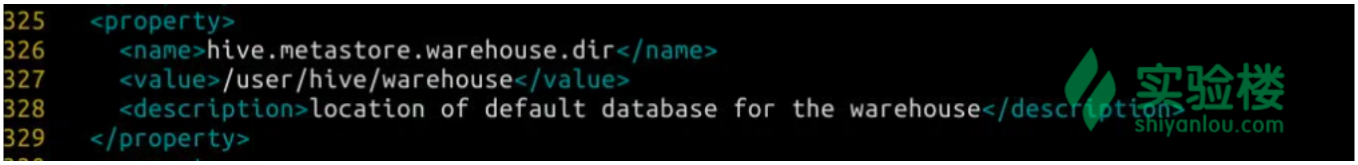
五、内嵌模式

（1）hive-site.xml

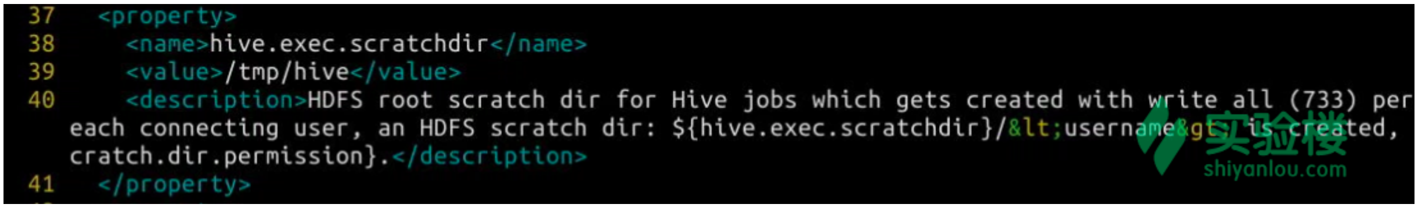
$HIVE_HOME/conf 对应的是 Hive 的配置文件路径，类似于之前学习的HBase, 该路径下的 hive-site.xml 是 Hive 工程的配置文件。默认情况下，该文件并不存在，我们需要拷贝它的模版来实现（这里暂时不需要修改，先拷贝）：

```
$ sudo cp hive-default.xml.template hive-site.xml
```

hive-site.xml 的主要配置有一下，不需要修改：



```
  <property>
    <name>hive.metastore.warehouse.dir</name>
    <value>/user/hive/warehouse</value>
    <description>location of default database for the warehouse</description>
  </property>
```



```
  <property>
    <name>hive.exec.scratchdir</name>
    <value>/tmp/hive</value>
    <description>HDFS root scratch dir for Hive jobs which gets created with write all (733) per
each connecting user, an HDFS scratch dir: ${hive.exec.scratchdir}/&lt;username&gt; is created,
cratch.dir.permission}.</description>
  </property>
```

- **hive.metastore.warehouse.dir**

  该参数指定了 Hive 的数据存储目录，默认位置在 HDFS 上面的 /user/hive/warehouse 路径下。

- **hive.exec.scratchdir**

  该参数指定了 Hive 的数据临时文件目录，默认位置为 HDFS 上面的 /tmp/hive 路径下。

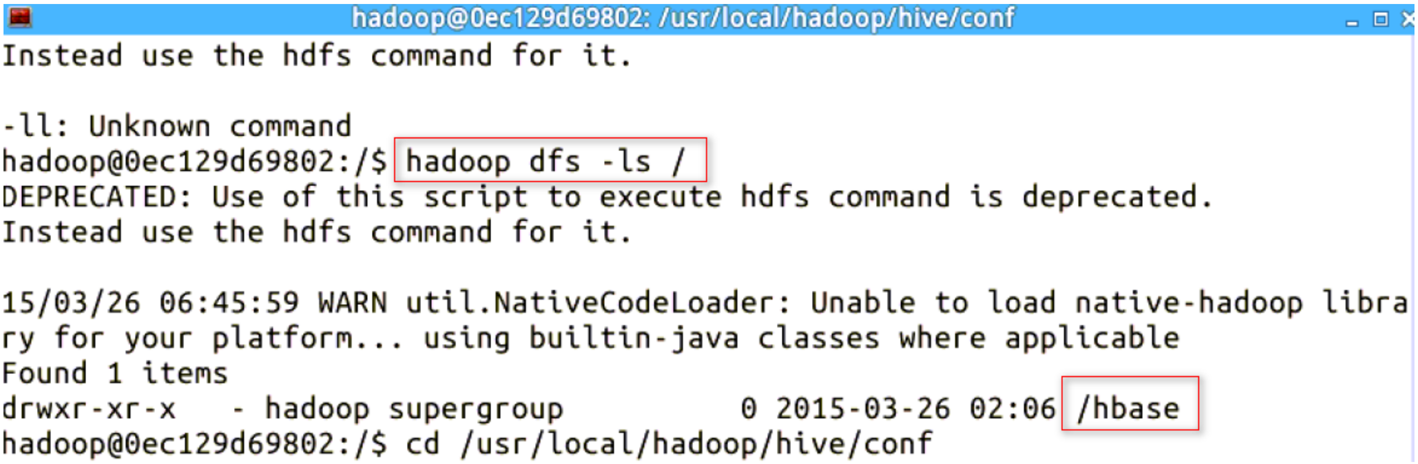
同时我们还要修改 Hive 目录下 /conf/hive-env.sh 文件（请根据自己的实际路径修改），该文件默认也不存在，同样是拷贝它的模版来修改：

```
export HADOOP_HEAPSIZE=1024

# Set HADOOP_HOME to point to a specific hadoop install directory
HADOOP_HOME=/opt/hadoop-2.4.1

# Hive Configuration Directory can be controlled by:
export HIVE_CONF_DIR=/opt/apache-hive-2.0.0-bin/conf

# Folder containing extra ibraries required for hive compilation/execution can be controlled by:
export HIVE_AUX_JARS_PATH=/opt/apache-hive-2.0.0-bin/lib
```

（2）创建必要目录

前面我们看到 hive-site.xml 文件中有两个重要的路径，切换到 hadoop 用户下查看 HDFS （需要先启动 Hadoop）是否有这些路径：

```
su hadoop                          # 进入hadoop用户，密码：hadoop
ssh localhost                      # 确保ssh直连
start-all.sh                       # 启动hadoop,确保已经执行bin/hdfs namenode -format,否则会报错
$ hadoop dfs -ls /
```



```
Instead use the hdfs command for it.

-ll: Unknown command
hadoop@0ec129d69802:/$ hadoop dfs -ls /
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.

15/03/26 06:45:59 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
Found 1 items
drwxr-xr-x   - hadoop supergroup          0 2015-03-26 02:06 /hbase
hadoop@0ec129d69802:/$ cd /usr/local/hadoop/hive/conf
```

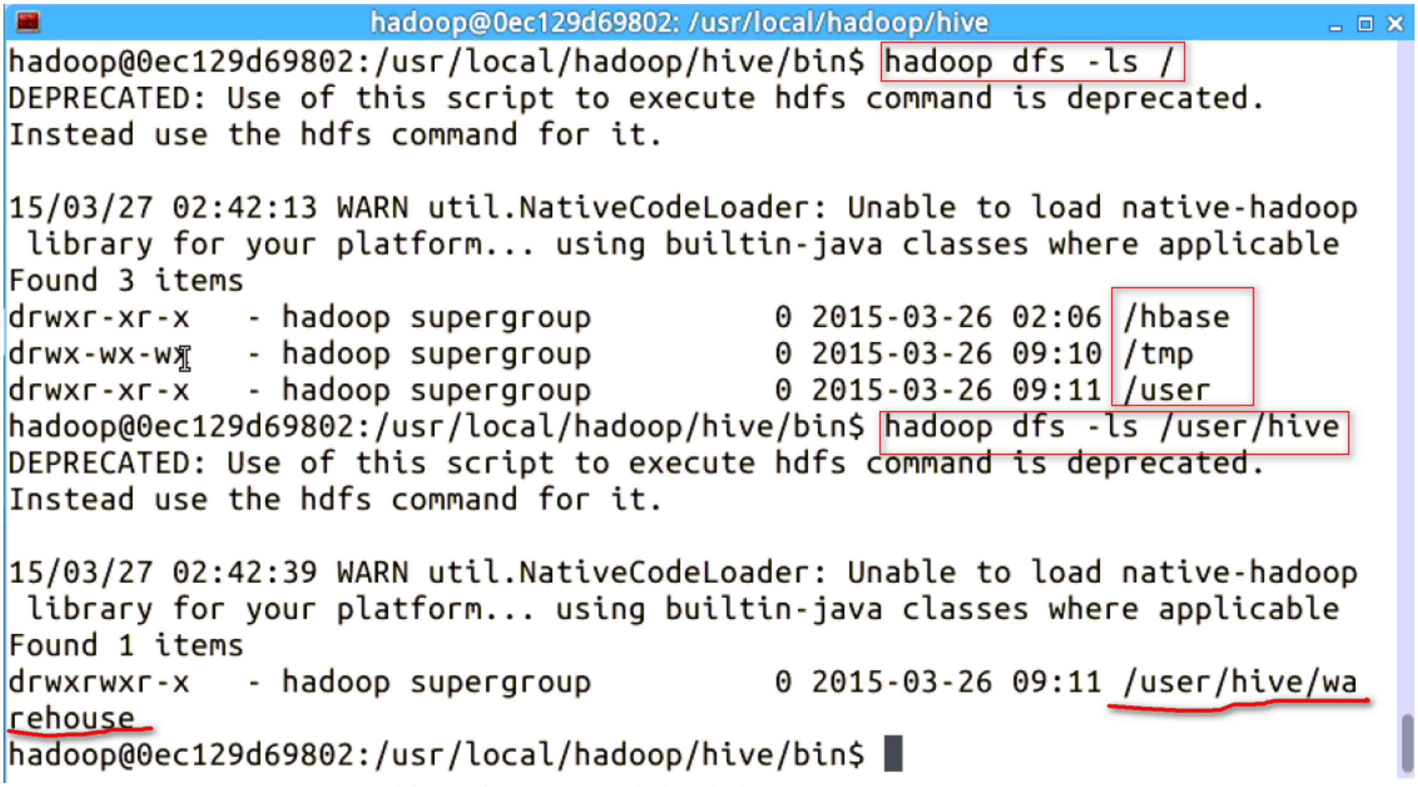
没有发现上面提到的路径，因此我们需要自己新建这些目录，并且给它们赋予用户写（W）权限。

```
$ hadoop dfs -mkdir /user/hive
$ hadoop dfs -mkdir /user/hive/warehouse
$ hadoop dfs -mkdir /tmp/hive
$ hadoop dfs -chmod 777 /user/hive/warehouse
$ hadoop dfs -chmod 777 /tmp/hive
```

如果你遇到 `no such file or directory` 类似的错误，就一步一步新建目录，例如：

```
$ hadoop dfs -mkdir /tmp
$ hadoop dfs -mkdir /tmp/hive
```

检查是否新建成功 `hadoop dfs -ls /` 以及 `hadoop dfs -ls /user/hive/` ：



```
hadoop@0ec129d69802:/usr/local/hadoop/hive/bin$ hadoop dfs -ls /
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.

15/03/27 02:42:13 WARN util.NativeCodeLoader: Unable to load native-hadoop
 library for your platform... using builtin-java classes where applicable
Found 3 items
drwxr-xr-x   - hadoop supergroup          0 2015-03-26 02:06 /hbase
drwx-wx-wx   - hadoop supergroup          0 2015-03-26 09:10 /tmp
drwxr-xr-x   - hadoop supergroup          0 2015-03-26 09:11 /user
hadoop@0ec129d69802:/usr/local/hadoop/hive/bin$ hadoop dfs -ls /user/hive
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.

15/03/27 02:42:39 WARN util.NativeCodeLoader: Unable to load native-hadoop
 library for your platform... using builtin-java classes where applicable
Found 1 items
drwxrwxr-x   - hadoop supergroup          0 2015-03-26 09:11 /user/hive/wa
rehouse
hadoop@0ec129d69802:/usr/local/hadoop/hive/bin$
```

（3）修改 io.tmpdir 路径

同时，要修改 hive-site.xml 中所有包含 `${system:java.io.tmpdir}` 字段的 value 即路径（vim下 / 表示搜索，后面跟你的关键词，比如搜索 hello，则为 `/hello`，再回车即可），你可以自己新建一个目录来替换它，例如 `/home/hive/iotmp` . 同样注意修改写权限。（以下是一个修改示例，其他的同理）

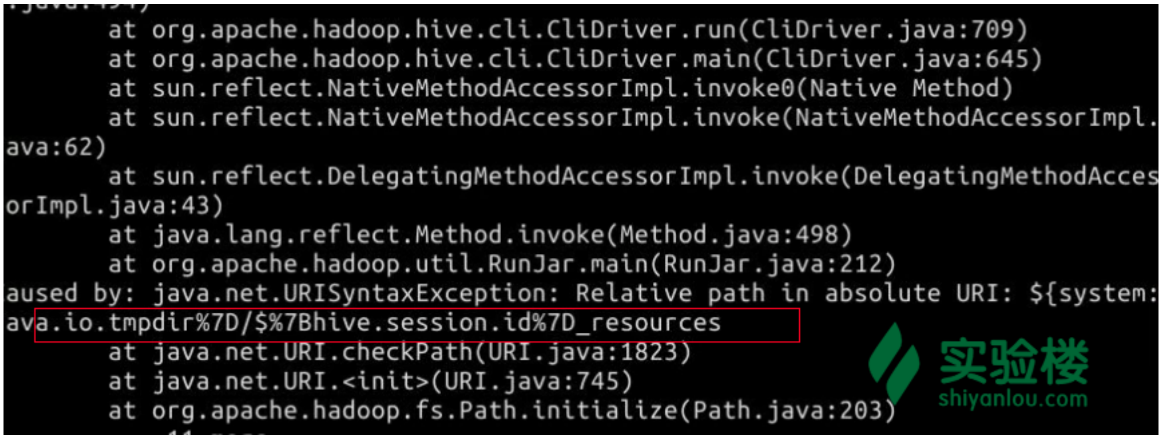
将

```
<value>${system:java.io.tmpdir}/${system:user.name}</value>
```

改成

```
<value>/home/hive/iotmp</value>
```

如果不修改这个目录，你很可能会出现如下错误：

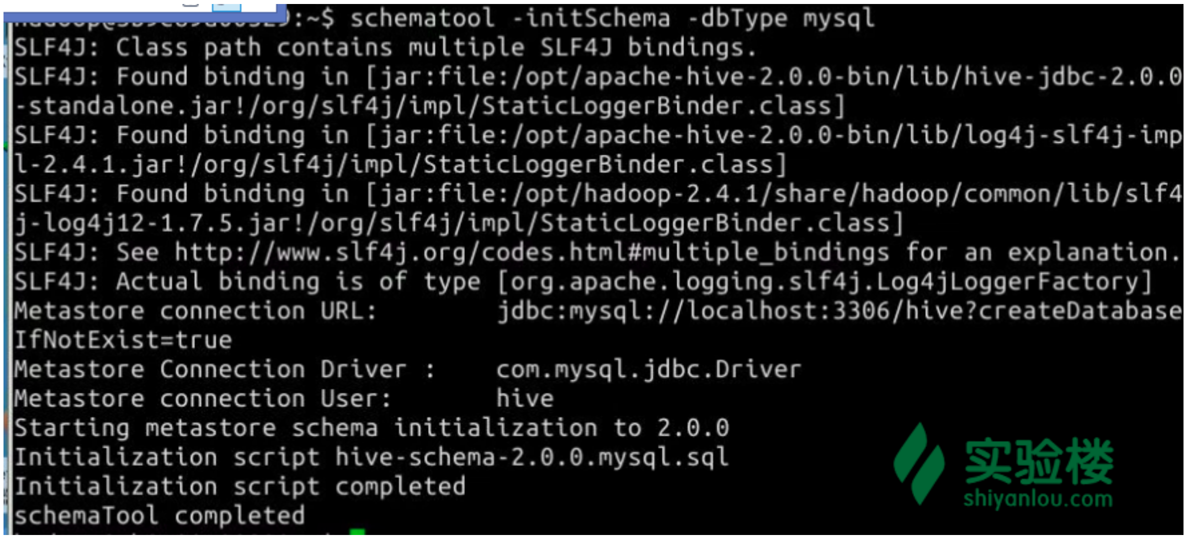


```
        at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:709)
        at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:645)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.
ava:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAcces
orImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:212)
aused by: java.net.URISyntaxException: Relative path in absolute URI: ${system:
ava.io.tmpdir%7D/$%7Bhive.session.id%7D_resources
        at java.net.URI.checkPath(URI.java:1823)
        at java.net.URI.<init>(URI.java:745)
        at org.apache.hadoop.fs.Path.initialize(Path.java:203)
```

（4）运行 Hive

前面我们已经提到过，内嵌模式使用默认配置和 Derby 数据库，所以无需其它特别修改直接运行 `hive`（确保 Hadoop 已经先启动）。完整步骤如下：

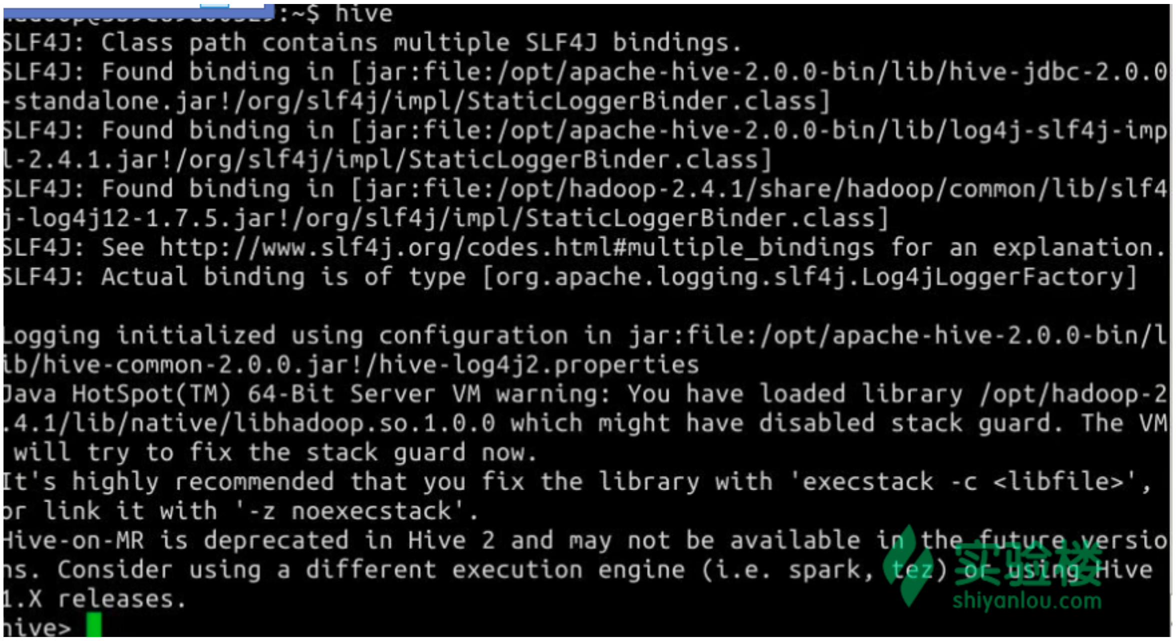
```
su hadoop                             # 进入hadoop用户，密码：hadoop
ssh localhost                         # 确保ssh直连
start-all.sh                          # 启动hadoop
schematool -dbType derby -initSchema  # 初始化hive的元数据库
```



```
:~$ schematool -initSchema -dbType mysql
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/apache-hive-2.0.0-bin/lib/hive-jdbc-2.0.0
-standalone.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/apache-hive-2.0.0-bin/lib/log4j-slf4j-imp
l-2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-2.4.1/share/hadoop/common/lib/slf4
j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Metastore connection URL:        jdbc:mysql://localhost:3306/hive?createDatabase
IfNotExist=true
Metastore Connection Driver :    com.mysql.jdbc.Driver
Metastore connection User:       hive
Starting metastore schema initialization to 2.0.0
Initialization script hive-schema-2.0.0.mysql.sql
Initialization script completed
schemaTool completed
```

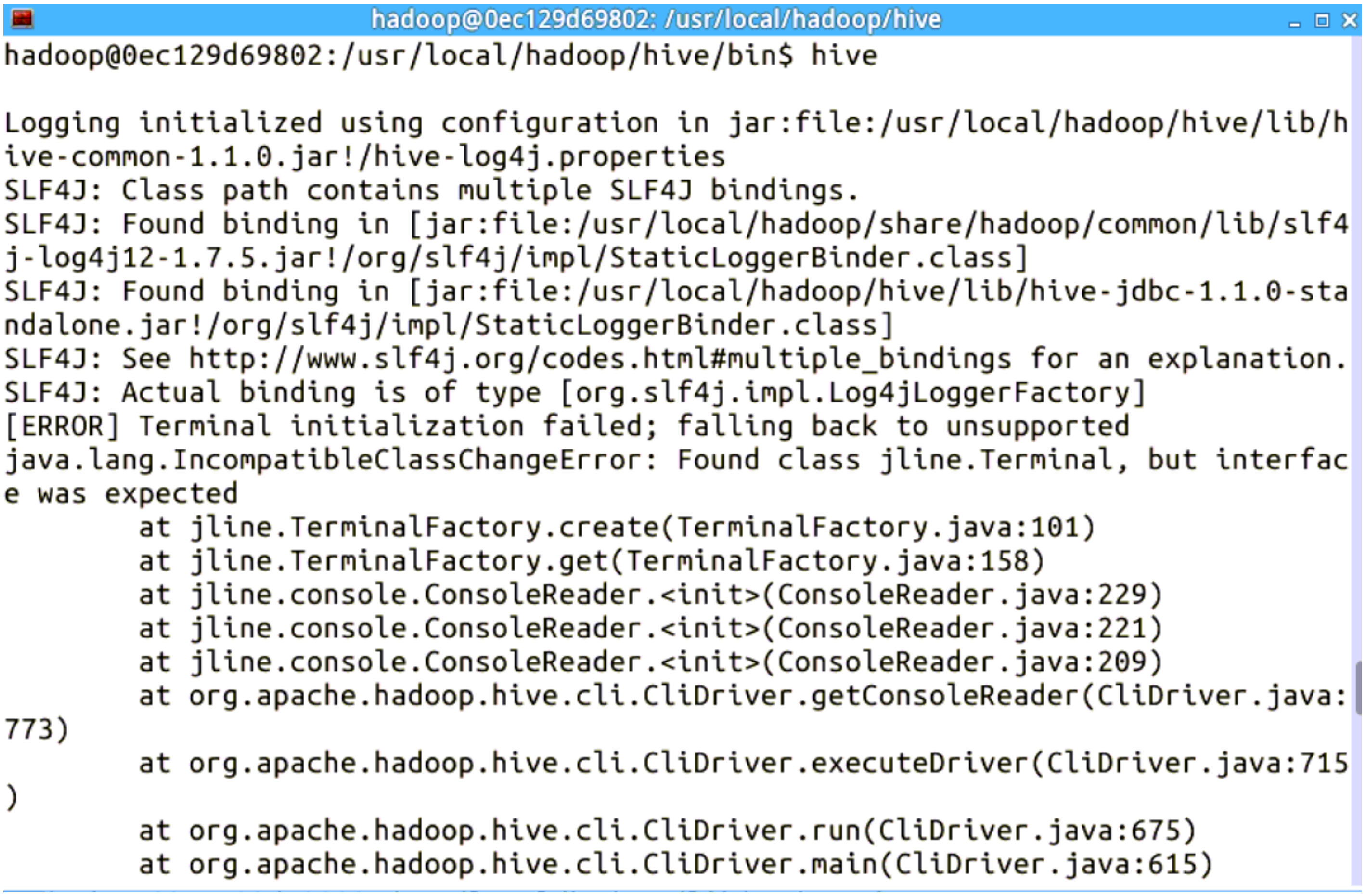
```
$ hive  #启动hive
```

动手实践是学习 IT 技术最有效的方式！　　**开始实验**



```
hadoop@[illegible]:~$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/apache-hive-2.0.0-bin/lib/hive-jdbc-2.0.0
-standalone.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/apache-hive-2.0.0-bin/lib/log4j-slf4j-imp
l-2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-2.4.1/share/hadoop/common/lib/slf4
j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/opt/apache-hive-2.0.0-bin/l
ib/hive-common-2.0.0.jar!/hive-log4j2.properties
Java HotSpot(TM) 64-Bit Server VM warning: You have loaded library /opt/hadoop-2
.4.1/lib/native/libhadoop.so.1.0.0 which might have disabled stack guard. The VM
 will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>',
or link it with '-z noexecstack'.
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versio
ns. Consider using a different execution engine (i.e. spark, tez) or using Hive
1.X releases.
hive>
```

你很可能会遇到这个错误：

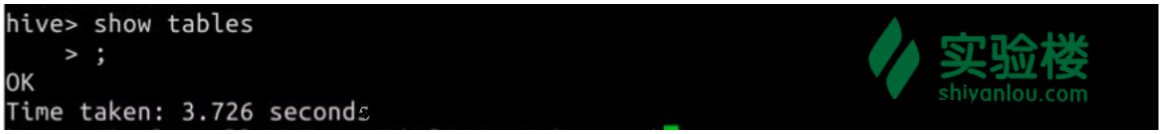


```
hadoop@0ec129d69802:/usr/local/hadoop/hive/bin$ hive

Logging initialized using configuration in jar:file:/usr/local/hadoop/hive/lib/h
ive-common-1.1.0.jar!/hive-log4j.properties
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4
j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop/hive/lib/hive-jdbc-1.1.0-sta
ndalone.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
[ERROR] Terminal initialization failed; falling back to unsupported
java.lang.IncompatibleClassChangeError: Found class jline.Terminal, but interfac
e was expected
        at jline.TerminalFactory.create(TerminalFactory.java:101)
        at jline.TerminalFactory.get(TerminalFactory.java:158)
        at jline.console.ConsoleReader.<init>(ConsoleReader.java:229)
        at jline.console.ConsoleReader.<init>(ConsoleReader.java:221)
        at jline.console.ConsoleReader.<init>(ConsoleReader.java:209)
        at org.apache.hadoop.hive.cli.CliDriver.getConsoleReader(CliDriver.java:
773)
        at org.apache.hadoop.hive.cli.CliDriver.executeDriver(CliDriver.java:715
)
        at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:675)
        at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:615)
```

这是因为 Hive 中的 Jline jar 包和 Hadoop 中的 Jline 冲突了，在路径：$HADOOP_HOME/share/hadoop/yarn/lib/jline-0.9.94.jar 将其删除。

再次启动 `hive` ，就OK了:

`show tables;` 注意不要漏写了 **分号**。



```
hive> show tables
    > ;
OK
Time taken: 3.726 seconds
```

六、本地模式

现在我们替换默认的 Derby 数据库为 MySQL数据库。

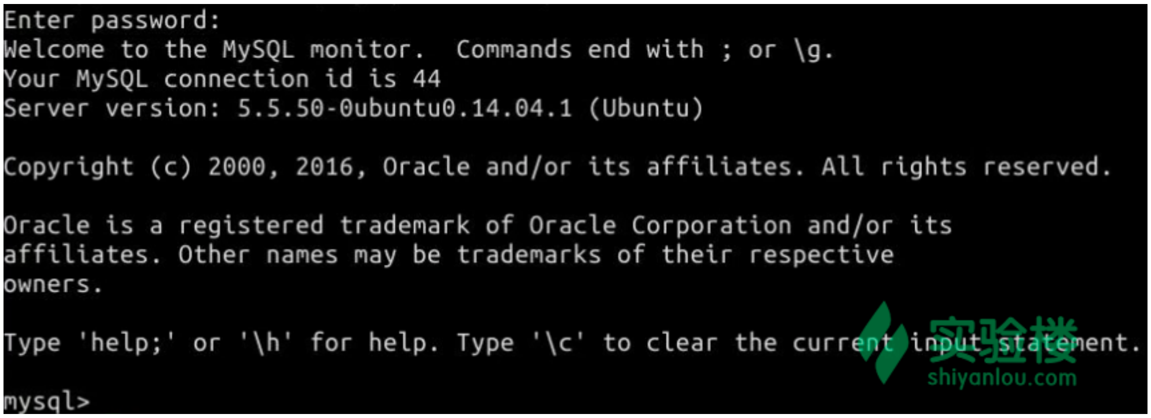
（1）下载安装 MySQL

```
$ sudo apt-get install mysql-server
```

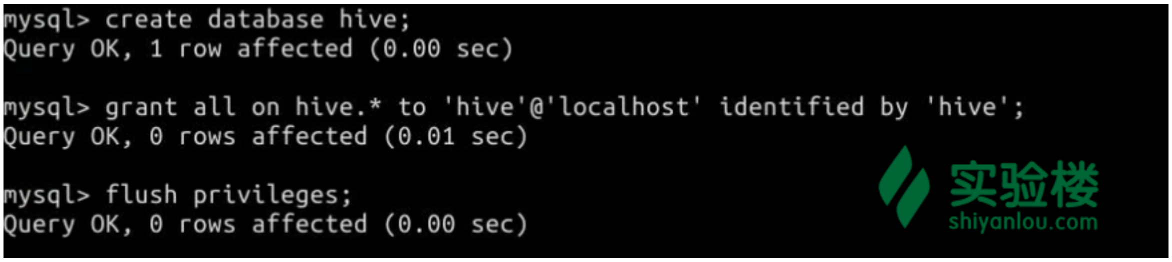
本实验环境下默认是安装了 MySQL 的，直接启动它：

```
$ sudo service mysql start
```

添加 root 用户，创建 hive 数据库：



```
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 44
Server version: 5.5.50-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```



```
mysql> create database hive;
Query OK, 1 row affected (0.00 sec)

mysql> grant all on hive.* to 'hive'@'localhost' identified by 'hive';
Query OK, 0 rows affected (0.01 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

虽然 MySQL 已经默认安装，但我们还需要下载一个 MySQL 的 JDBC 驱动包。这里使用的是 `mysql-connector-java-5.1.35-bin.jar` ，你需要将其复制到 $HIVE_HOME/lib 目录下面：

```
$ sudo mv  /home/shiyanlou/mysql-connector-java-5.1.35-bin.jar /usr/local/hadoop/hive/lib/
```

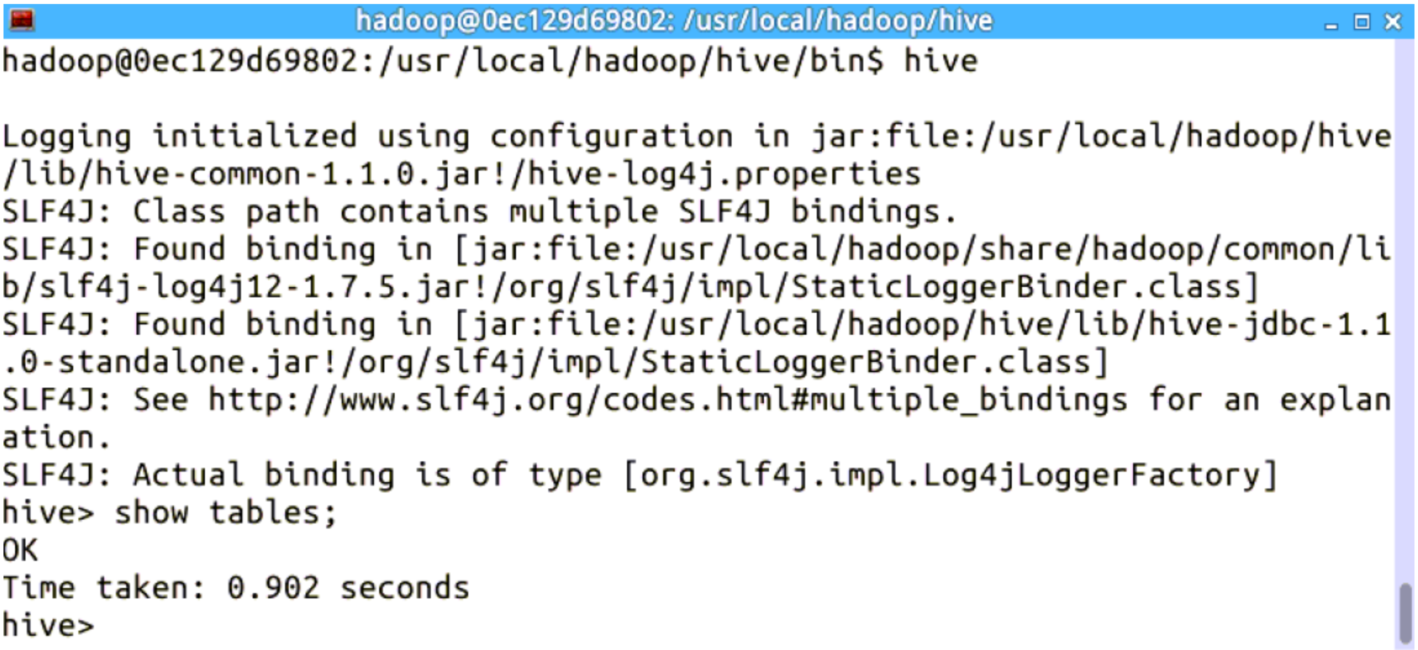
（2）修改 hive-site.xml 配置文件

最后，依然是修改 $HIVE_HOME/conf 下的 hive-site.xml 文件，把默认的 Derby 修改为 MySQL：

```
<property>
    <name>javax.jdo.option.ConnectionURL</name>
    //所连接的MySQL数据库实例
    <value>jdbc:mysql://localhost:3306/hive?createDatabaseIfNotExist=true</value>
</property>

<property>
    <name>javax.jdo.option.ConnectionDriverName</name>
    //连接的MySQL数据库驱动
    <value>com.mysql.jdbc.Driver</value>
</property>

<property>
    <name>javax.jdo.option.ConnectionUserName</name>
    //连接的MySQL数据库用户名
    <value>hive</value>
</property>

<property>
    <name>javax.jdo.option.ConnectionPassword</name>
    //连接的MySQL数据库密码
    <value>hive</value>
</property>
```

（3）启动 Hive

启动 Hive 的方式同内嵌模式一样：



```
hadoop@0ec129d69802:/usr/local/hadoop/hive/bin$ hive

Logging initialized using configuration in jar:file:/usr/local/hadoop/hive
/lib/hive-common-1.1.0.jar!/hive-log4j.properties
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/li
b/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop/hive/lib/hive-jdbc-1.1
.0-standalone.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explan
ation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
hive> show tables;
OK
Time taken: 0.902 seconds
hive>
```

七、课后作业

有条件的同学，可以自行尝试安装配置远程模式下的 Hive。

八、参考文档

- 《Hadoop实战 第2版》陆嘉恒，机械工业出版社；
- 【Hive一】Hive入门 (http://bit1129.iteye.com/blog/2169918)；
- Hive安装配置 (http://hatech.blog.51cto.com/8360868/1427748)；

上一节 (/courses/38/labs/760/document)

下一节 (/courses/38/labs/772/document)

课程教师