**安装及配置Hive**

# 一、实验目的

* 熟悉Hive
* 掌握Hive的基本配置

# 二、实验内容

本实验主要是为了在hadoop-master节点上安装并配置Hive。Apache Hive是基于Hadoop的一个数据仓库工具，可以将结构化的数据文件映射为一张数据库表，并提供一种HQL语言进行查询，具有扩展性好、延展性好、高容错等特点，多应用于离线数仓建设。通过本实验，学生可以增加对Hive的了解，为后续的大数据处理做准备。

# 三、实验知识点

* Hive的安装和配置
* Hive的基本操作

# 四、实验环境

* Centos 7.9.2009 64位
* Hive 3.1.3

# 五、实验步骤

## 1. 安装Hive

首先，通过链接<https://dlcdn.apache.org/hive/hive-3.1.3/apache-hive-3.1.3-bin.tar.gz>下载hive。

（1）安装Hive

***cd /home***

***mkdir hive***

然后，将编译好的apache-hive-3.1.3-bin.tar.gz复制到/home/hive目录下。执行如下命令：

***tar -zxvf apache-hive-3.1.3-bin.tar.gz***

这样Hive就安装完成了。

## 2. 配置Hive

### 2.1 环境准备

（1）创建hive数据库

首先，登录MariaDB创建数据库“hive”。

***mysql -u root -p***

***create database hive DEFAULT CHARSET utf8mb4 COLLATE utf8mb4\_unicode\_ci;***

然后，创建hive数据库对应的用户，并授予访问权限。

***grant all on hive.\* to 'hive'@'%' IDENTIFIED BY 'hivepwd';***

***grant all on hive.\* to 'hive'@'hadoop-master' IDENTIFIED BY 'hivepwd';***

***grant all on hive.\* to 'hive'@'localhost' IDENTIFIED BY 'hivepwd';***

***grant all privileges on \*.\* to 'hive'@'%' identified by 'hivepwd';***

***grant all privileges on \*.\* to 'hive'@'hadoop-master' identified by 'hivepwd';***

***grant all privileges on \*.\* to 'hive'@'localhost' identified by 'hivepwd';***

***flush privileges;***

***exit;***

（2）环境变量配置

使用命令***vim /etc/profile***，编辑/etc/profile，在文件的最后添加如下内容：

***export HIVE\_HOME=/home/hive/apache-hive-3.1.3-bin***

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

然后，使用***source /etc/profile***命令使环境变量的设置生效。

（3）下载mysql驱动程序

如果Hive使用MySql作为元数据存储方式，则需要添加MySql的驱动程序到Hive的lib目录下。因此，需要下载mysql-connector-java-5.1.48-bin.jar文件，并将文件复制到/home/hive/ apache-hive-3.1.3-bin/lib目录下。具体执行命令如下：

***cd ~***

***wget https://cdn.mysql.com/archives/mysql-connector-java-5.1/mysql-connector-java-5.1.48.tar.gz***

***tar zxvf mysql-connector-java-5.1.48.tar.gz***

***cd mysql-connector-java-5.1.48***

***cp mysql-connector-java-5.1.48-bin.jar /home/hive/apache-hive-3.1.3-bin/lib***

***cd ~***

### 2.2 修改Hive相关配置

（1）修改hive-env.sh和hive-site.xml文件

首先，进入/home/hive/apache-hive-3.1.3-bin/conf目录，根据模板创建hive-env.sh和hive-site.xml文件。具体执行如下命令：

<property>

<name>system:java.io.tmpdir</name>

<value>/tmp/hive/java</value>

</property>

<property>

<name>system:user.name</name>

<value>${user.name}</value>

</property>

***cd /home/hive/apache-hive-3.1.3-bin/conf***

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

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

然后，修改hive-env.sh文件，文件位于/home/hive/apache-hive-3.1.3-bin/conf目录下，添加如下内容：

***HADOOP\_HOME=/home/hadoop/hadoop-3.3.3***

***export HIVE\_CONF\_DIR=/home/hive/apache-hive-3.1.3-bin/conf***

（2）修改hive-site.xml文件

找到hive-site.xml文件，文件位于/home/hive/apache-hive-3.1.3-bin/conf目录下，注意需要将<configuration>标签中的内容全部清空，然后添加如下内容：

***<property>***

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

***<value>hive</value>***

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

***</property>***

***<property>***

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

***<value>hivepwd</value>***

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

***</property>***

***<property>***

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

***<value>jdbc:mysql://hadoop-master:3306/hive?createDatabaseIfNotExist=true</value>***

***<description>***

***JDBC connect string for a JDBC metastore.***

***To use SSL to encrypt/authenticate the connection, provide database-specific SSL flag in the connection URL.***

***For example, jdbc:postgresql://myhost/db?ssl=true for postgres database.***

***</description>***

***</property>***

***<property>***

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

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

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

***</property>***

***<property>***

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

***<value>/home/hive/apache-hive-3.1.3-bin/iotmp</value>***

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

***</property>***

***<property>***

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

***<value>/home/hive/apache-hive-3.1.3-bin/iotmp</value>***

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

***</property>***

然后，根据hive-site.xml的配置，创建缓存目录：

***cd /home/hive/apache-hive-3.1.3-bin***

***mkdir iotmp***

（3）修改hive-config.sh文件

找到hive-config.sh文件，文件位于/home/hive/apache-hive-3.1.3-bin/bin目录下，在文件的最开始添加如下内容：

***export JAVA\_HOME=/usr/lib/jvm/java-1.8.0***

***export HIVE\_HOME=/home/hive/apache-hive-3.1.3-bin***

***export HADOOP\_HOME=/home/hadoop/hadoop-3.3.3***

（4）替换不一致的guava库

执行如下命令，替换guava。具体命令如下：

***rm /home/hive/apache-hive-3.1.3-bin/lib/guava-19.0.jar***

***cd /home/hadoop/hadoop-3.3.3/share/hadoop/hdfs/lib/***

***cp guava-27.0-jre.jar /home/hive/apache-hive-3.1.3-bin/lib/***

## 2.3 配置slave节点为hive客户端

如果不在master节点上访问Hive，则需要配置slave节点为客户端。

（1）复制hive到slave节点

首先，从master节点上分发hive到slave节点上。具体执行命令如下：

***cd /home/hive***

***tar -zcvf apache-hive-3.1.3-bin.tar.gz ./apache-hive-3.1.3-bin/***

***scp apache-hive-3.1.3-bin.tar.gz root@hadoop-slave1:/root/***

***rm -rf apache-hive-3.1.3-bin.tar.gz***

（2）在slave节点上配置hive

首先，在hadoop-master节点上，输入如下命令登录hadoop-slave1登录slave1节点：

***ssh root@hadoop-slave1***

然后，在slave1节点上配置hive，具体执行如下命令：

***cd /home***

***mkdir hive***

***cd hive***

***cp /root/apache-hive-3.1.3-bin.tar.gz .***

***tar zxvf apache-hive-3.1.3-bin.tar.gz***

***cd apache-hive-3.1.3-bin/conf***

修改hive-site.xml文件，清空<configuration>标签中的内容，增加内容如下：

***<property>***

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

***<value>thrift://hadoop-master:9083</value>***

***<description>Thrift URI for the remote metastore. Used by metastore client to connect to remote metastore.</description>***

***</property>***

如果还想在slave2上配置hive客户端的话，重复前面的步骤，并将slave1更换为slave2。

## 2.4 启动Hive

（1）首先启动hadoop

启动hive之前需要先启动hadoop。执行start-all.sh启动hadoop。

（2）初次启动初始化数据库

在初次启动hive的时候需要初始化数据，以后就不需要了。执行如下命令进行初始化：

***schematool -initSchema -dbType mysql***

（4）启动metastore服务

执行如下命令：

***cd ~***

***nohup hive --service metastore &***

启动的日志信息将在~/nohup.out下，可以通过查看日志确定是否启动成功。

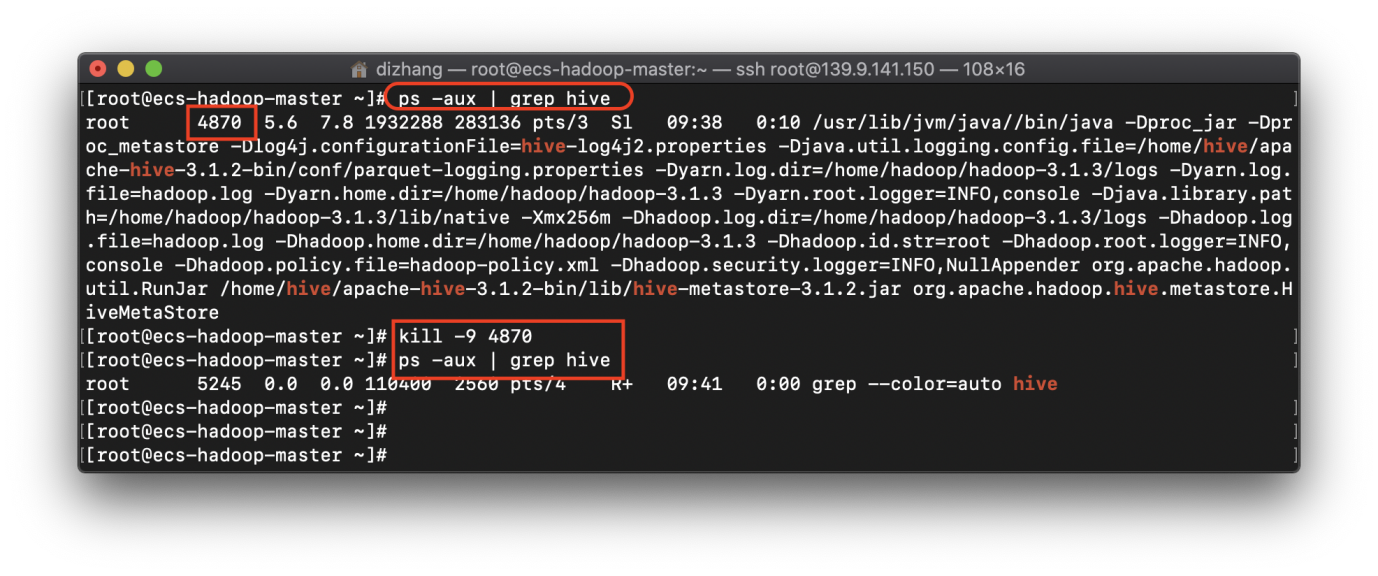
（5）停止hive的metastore服务

如果需要关闭metastore服务的话，通过如下命令可以查看metastore服务的进程号，然后通过kill命令杀死对应的进程即可关闭服务，具体操作如下：

***ps -aux | grep hive***

***kill -9 4870***

执行kill命令时，需要替换成对应的进程号。效果如下图所示：



## 2.5 测试Hive

（1）在Hive服务端（hadoop-master）访问

执行hive命令，然后进行数据的创建等操作；具体操作内容和结果如下：

***$ hive***

***hive> create database src;***

***OK***

***Time taken: 0.215 seconds***

***hive> show databases;***

***OK***

***default***

***src***

***Time taken: 0.04 seconds, Fetched: 2 row(s)***

***hive> use src;***

***OK***

***Time taken: 0.055 seconds***

***hive> create table test1(id int);***

***OK***

***Time taken: 0.692 seconds***

***hive> show tables;***

***OK***

***test1***

***Time taken: 0.045 seconds, Fetched: 1 row(s)***

（2）在Hive客户端（hadoop-slave1）访问

访问前需要保证metastore服务已经启动，也即在hadoop-master上执行了如下命令：

***cd ~***

***nohup hive --service metastore &***

另外，在slave1节点上也需要按照2.1步骤（2）配置环境变量。然后输入hive命令执行。具体操作内容和结果如下：

***$ hive***

***hive> show databases;***

***OK***

***default***

***src***

***Time taken: 0.692 seconds, Fetched: 2 row(s)***

***hive> use src;***

***OK***

***Time taken: 0.05 seconds***

***hive> show tables;***

***OK***

***test1***

***Time taken: 0.055 seconds, Fetched: 1 row(s)***

***hive> create table test2(id int, name string);***

***OK***

***Time taken: 1.795 seconds***

***hive> show tables;***

***OK***

***test1***

***test2***

***Time taken: 0.059 seconds, Fetched: 2 row(s)***

***hive>***

## 2.6 其他注意事项

（1）启动hive之前一定要先启动hadoop；

（2）通过hive所创建的数据库及其表都是默认存放在hdfs的 /user/hive/warehouse 目录底下。我们也可以通过修改hive-site.xml文件更改，例

<property>

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

<value>hdfs://hadoop-master:9000/hive/warehouse</value>

</property>

通过Browse the file system可以看到src.db文件。效果如下图所示。

