**Hive安装教程**

# 上传解压压缩文件

tar -zxvf apache-hive-0.13.1-bin.tar.gz

# 配置配置文件

## 重命名配置文件

将/home/hadoop/hive-0.13.1/conf下的hive-env.sh.template重命名为hive-env.sh

## 配置hive-env.sh文件

HADOOP\_HOME=/home/hadoop/hadoop-2.5.0

export HIVE\_CONF\_DIR=/home/hadoop/hive-0.13.1/conf

## 创建hdfs目录

hdfs dfs -mkdir /tmp

hdfs dfs -mkdir -p /user/hive/warehouse

## 配置MySQL存储元数据

新建hive-site.xml文件

<?xml version="1.0"?>

<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>

<property>

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

<value>

jdbc:mysql://MyDream:3306/hive?createDatabaseIfNotExist=true

</value>

<description>JDBC connect string for a JDBC metastore</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>javax.jdo.option.ConnectionUserName</name>

<value>root</value>

<description>username to use against metastore database

</description>

</property>

<property>

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

<value>123456</value>

<description>password to use against metastore database

</description>

</property>

</configuration>

## 上传jar包

上传mysql-connector-java-5.1.41-bin.jar到/home/hadoop/hive-0.13.1/lib目录

## 配置/etc/profile

export HIVE\_HOME=/home/hadoop/hive-0.13.1

export MYSQL\_HOME=/usr/local/mysql

export PATH=$PATH:$JAVA\_HOME/bin:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin:$ZOOKEEPER\_HOME/bin:$MYSQL\_HOME/bin:$HIVE\_HOME/bin

# hive日志查看

## 配置日志输出目录

将/home/hadoop/hive-0.13.1/conf下的hive-log4j.properties.template

重命名为hive-log4j.properties

本文件中有以下行：

hive.log.threshold=ALL

hive.root.logger=INFO,DRFA

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

hive.log.file=hive.log

日志文件在${java.io.tmpdir}/${user.name}(系统tmp目录下的用户名目录下)目录下，即/tmp/hadoop/hive.log

也可更改配置到自己定义的目录

hive.root.logger=INFO,DRFA 日志级别

hive.log.dir= /home/hadoop/hive-0.13.1/tmp 日志目录

# 显示信息

在cli命令行上显示当前数据库，以及查询表的行头信息

$HIVE\_HOME/conf/hive-site.xml

<property>

<name>hive.cli.print.header</name>

<value>true</value>

<description>Whether to print the names of the columns in query output.

</description>

</property>

<property>

<name>hive.cli.print.current.db</name>

<value>true</value>

<description>Whether to include the current database in the

Hive prompt.</description>

</property>

# Hive服务用户权限

## 开启用户验证

### 配置hive-site.xml

--开启权限验证

--表的创建者对表拥有所有权限

<property>

<name>hive.security.authorization.enabled</name>

<value>true</value>

<description>enable or disable the hive client authorization</description>

</property>

<property>

<name>hive.security.authorization.createtable.owner.grants</name>

<value>ALL</value>

<description>the privileges automatically granted to the owner whenever a table gets created. An example like "select,drop" will grant select and drop privilege to the owner of the table</description>

</property>

# [HiveServer2用户名密码验证](http://blog.csdn.net/hua_ed/article/details/51693659)

## Hive-site.xml添加配置

--设置server2验证方式是CUSTOM

并指定自定义验证类是org.apache.hadoop.hive.contrib.auth.ZKJPasswdAuthenticator

1. **<property>**
2. **<name>**hive.server2.authentication**</name>**
3. **<value>**CUSTOM**</value>**
4. **</property>**
6. **<property>**
7. **<name>**hive.server2.custom.authentication.class**</name>**
8. **<value>**org.apache.hadoop.hive.contrib.auth.ZKJPasswdAuthenticator**</value>**
9. **</property>**