目录

[1. 实验一：Hive组件安装配置](#_Toc675)

[1.1. 实验目的](#_Toc25670)

[1.2. 实验要求](#_Toc14570)

[1.3. 实验环境](#_Toc2948)

[1.4. 实验过程](#_Toc10896)

[1.4.1. 实验任务一：下载和解压安装文件](#_Toc12176)

[1.4.2. 实验任务二：设置Hive环境](#_Toc17944)

# 实验一：Hive组件安装配置

## 实验目的

完成本实验，您应该能够：

* 掌握Hive组件安装配置
* 掌握Hive组件格式化和启动

## 实验要求

* 熟悉Hive组件安装配置
* 了解Hive组件格式化和启动

## 实验环境

本实验所需之主要资源环境如表1-1所示。

表1-1 资源环境

|  |  |
| --- | --- |
| **服务器集群** | 3个以上节点，节点间网络互通，各节点最低配置：双核CPU、8GB内存、100G硬盘 |
| **运行环境** | CentOS 7.4 |
| **大数据平台** | H3C教学与实践管理系统 |
| **服务和组件** | 完成前面章节实验，其他服务及组件根据实验需求安装 |

## 实验过程

### 实验任务一：下载和解压安装文件

#### 步骤一：基础环境和安装准备

Hive组件需要基于Hadoop系统进行安装。因此，在安装Hive组件前，需要确保Hadoop系统能够正常运行。本章节内容是基于之前已部署完毕的Hadoop全分布系统，在master节点上实现Hive组件安装。

Hive组件的部署规划和软件包路径如下：

（1）当前环境中已安装Hadoop全分布系统。

（2）本地安装MySQL数据库（账号root，密码Password123$），

软件包在/opt/software/mysql-5.7.18路径下。

（3）MySQL端口号（3306）。

（4）MySQL的JDBC驱动包/opt/software/mysql-connector-java-5.1.47.jar，

在此基础上更新Hive元数据存储。

（5）Hive软件包/opt/software/apache-hive-2.0.0-bin.tar.gz。

#### 步骤二：解压安装文件

（1）使用root用户，将Hive安装包

/opt/software/apache-hive-2.0.0-bin.tar.gz路解压到/usr/local/src路径下。

[root@master ~]# tar -zxvf /opt/software/apache-hive-2.0.0-bin.tar.gz -C /usr/local/src

（2）将解压后的apache-hive-2.0.0-bin文件夹更名为hive；

[root@master ~]# mv /usr/local/src/apache-hive-2.0.0-bin /usr/local/src/hive

（3）修改hive目录归属用户和用户组为hadoop

[root@master ~]# chown -R hadoop:hadoop /usr/local/src/hive

### 实验任务二：设置Hive环境

#### 步骤一：卸载MariaDB数据库

Hive元数据存储在MySQL数据库中，因此在部署Hive组件前需要首先在Linux系统下安装MySQL数据库，并进行MySQL字符集、安全初始化、远程访问权限等相关配置。需要使用root用户登录，执行如下操作步骤：

（1）关闭Linux系统防火墙，并将防火墙设定为系统开机并不自动启动。

# 关闭防火墙服务

[root@master ~]# systemctl stop firewalld

# 设置防火墙服务开机不启动

[root@master ~]# systemctl disable firewalld

（2）卸载Linux系统自带的MariaDB。

1）首先查看Linux系统中MariaDB的安装情况。

# 查询已安装的mariadb软件包

[root@ master ~]# rpm -qa | grep mariadb

**mariadb-libs-5.5.52-2.el7.x86\_64**

以上结果显示Linux系统中已经按照了mariadb-libs-5.5.52-2.el7.x86\_64软件包，需要将其卸载。

2）卸载MariaDB软件包。

# 卸载mariadb软件包

[root@master ~]# rpm -e --nodeps mariadb-libs-5.5.56-2.el7.x86\_64

#### 步骤二：安装MySQL数据库

（1）按如下顺序依次按照MySQL数据库的mysql common、mysql libs、mysql client软件包。

# MySQL软件包路径

[root@master ~]# cd /opt/software/mysql-5.7.18/

[root@master mysql-5.7.18]# rpm -ivh mysql-community-common-5.7.18-1.el7.x86\_64.rpm

[root@master mysql-5.7.18]# rpm -ivh mysql-community-libs-5.7.18-1.el7.x86\_64.rpm

[root@master mysql-5.7.18]# rpm -ivh mysql-community-client-5.7.18-1.el7.x86\_64.rpm

（2）安装mysql server软件包。

[root@master mysql-5.7.18]# rpm -ivh mysql-community-server-5.7.18-1.el7.x86\_64.rpm

[root@master mysql-5.7.18]# cd ~

（3）修改MySQL数据库配置，在/etc/my.cnf文件中添加如表6-1所示的MySQL数据库配置项。

表6-1 MySQL数据库配置

|  |  |  |
| --- | --- | --- |
| 字段 | 配置值 | 字段说明 |
| default-storage-engine | innodb | 设置innodb为默认的存储引擎。 |
| innodb\_file\_per\_table | - | 设置每个表的数据单独保存，而不是统一保存在innodb系统表空间中。单独保存有方便管理和提升性能两方面优势。 |
| collation-server | utf8\_general\_ci | 设置支持中文编码字符集。 |
| init-connect | 'SET NAMES utf8' | 设置用户登录到数据库之后，在执行第一次查询之前执行SET NAMES utf8命令，将使用的字符编码设定为utf8。 |
| character-set-server | utf8 | 将MySQL服务器字符集设定为utf8。 |

将以下配置信息添加到/etc/my.cnf文件symbolic-links=0配置信息的下方(文件倒数第三行)。

[root@master mysql-5.7.18]# vi /etc/my.cnf

default-storage-engine=innodb

innodb\_file\_per\_table

collation-server=utf8\_general\_ci

init-connect='SET NAMES utf8'

character-set-server=utf8

（4）启动MySQL数据库。

[root@master ~]# systemctl start mysqld

（5）查询MySQL数据库状态。mysqld进程状态为active (running)，则表示MySQL数据库正常运行。

如果mysqld进程状态为failed，则表示MySQL数据库启动异常。此时需要排查/etc/my.cnf文件。

[root@master ~]# systemctl status mysqld

**● mysqld.service - MySQL Server**

**Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; vendor preset: disabled)**

**Active: active (running) since 一 2020-05-11 16:28:27 CST; 1h 26min ago**

**Docs: man:mysqld(8)**

**http://dev.mysql.com/doc/refman/en/using-systemd.html**

**Process: 941 ExecStart=/usr/sbin/mysqld --daemonize --pid-file=/var/run/mysqld/mysqld.pid $MYSQLD\_OPTS (code=exited, status=0/SUCCESS)**

**Process: 911 ExecStartPre=/usr/bin/mysqld\_pre\_systemd (code=exited, status=0/SUCCESS)**

**Main PID: 944 (mysqld)**

**CGroup: /system.slice/mysqld.service**

**└─944 /usr/sbin/mysqld --daemonize --pid-file=/var/run/mysqld/mysqld.pid**

（6）查询MySQL数据库默认密码。

MySQL数据库安装后的默认密码保存在/var/log/mysqld.log文件中，在该文件中以password关键字搜索默认密码。

[root@master ~]# cat /var/log/mysqld.log | grep password

**2020-05-07T02:34:03.336724Z 1 [Note] A temporary password is generated for root@localhost: MPg5lhk4?>Ui #默认密码为MPg5lhk4?>Ui**

MySQL数据库是安装后随机生成的，所以每次安装后生成的默认密码不相同。

（7）MySQL数据库初始化。

执行mysql\_secure\_installation命令初始化MySQL数据库，初始化过程中需要设定数据库root用户登录密码，密码需符合安全规则，包括大小写字符、数字和特殊符号，可设定密码为Password123$。

在进行MySQL数据库初始化过程中会出现以下交互确认信息：

1）Change the password for root ? ((Press y|Y for Yes, any other key for No)表示是否更改root用户密码，在键盘输入y和回车。

2）Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for No)表示是否使用设定的密码继续，在键盘输入y和回车。

3）Remove anonymous users? (Press y|Y for Yes, any other key for No)表示是否删除匿名用户，在键盘输入y和回车。

4）Disallow root login remotely? (Press y|Y for Yes, any other key for No)表示是否拒绝root用户远程登录，在键盘输入n和回车，表示允许root用户远程登录。

5）Remove test database and access to it? (Press y|Y for Yes, any other key for No)表示是否删除测试数据库，在键盘输入y和回车。

6）Reload privilege tables now? (Press y|Y for Yes, any other key for No) 表示是否重新加载授权表，在键盘输入y和回车。

mysql\_secure\_installation命令执行过程如下：

[root@master ~]# mysql\_secure\_installation

**Securing the MySQL server deployment.**

**Enter password for user root:**  # 输入/var/log/mysqld.log文件中查询到的默认root用户登录密码

**The existing password for the user account root has expired. Please set a new password.**

**New password:** # 输入新密码Password123$

**Re-enter new password:** # 再次输入新密码Password123$

**The 'validate\_password' plugin is installed on the server.**

**The subsequent steps will run with the existing configuration**

**of the plugin.**

**Using existing password for root.**

**Estimated strength of the password: 100**

**Change the password for root ? ((Press y|Y for Yes, any other key for No) :** y #输入y

**New password:**  # 输入新密码Password123$

**Re-enter new password:** # 再次输入新密码Password123$

**Estimated strength of the password: 100**

**Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for No) : y**  # 输入y

**By default, a MySQL installation has an anonymous user,**

**allowing anyone to log into MySQL without having to have**

**a user account created for them. This is intended only for**

**testing, and to make the installation go a bit smoother.**

**You should remove them before moving into a production**

**environment.**

**Remove anonymous users? (Press y|Y for Yes, any other key for No) : y** # 输入y

**Success.**

**Normally, root should only be allowed to connect from**

**'localhost'. This ensures that someone cannot guess at**

**the root password from the network.**

**Disallow root login remotely? (Press y|Y for Yes, any other key for No) : n** # 输入n

**... skipping.**

**By default, MySQL comes with a database named 'test' that**

**anyone can access. This is also intended only for testing,**

**and should be removed before moving into a production**

**environment.**

**Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y** # 输入y

**- Dropping test database...**

**Success.**

**- Removing privileges on test database...**

**Success.**

**Reloading the privilege tables will ensure that all changes**

**made so far will take effect immediately.**

**Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y** # 输入y

**Success.**

**All done!**

（8）添加root用户从本地和远程访问MySQL数据库表单的授权。

[root@master ~]# mysql -uroot -p

**Enter password:**  # 输入新设定的密码Password123$

**Welcome to the MySQL monitor. Commands end with ; or \g.**

**Your MySQL connection id is 20**

**Server version: 5.7.18 MySQL Community Server (GPL)**

**Copyright (c) 2000, 2017, 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> grant all privileges on \*.\* to root@'localhost' identified by 'Password123$'; # 添加root用户本地访问授权

**Query OK, 0 rows affected, 1 warning (0.01 sec)**

mysql> grant all privileges on \*.\* to root@'%' identified by 'Password123$'; # 添加root用户远程访问授权

**Query OK, 0 rows affected, 1 warning (0.00 sec)**

mysql> flush privileges; # 刷新授权

**Query OK, 0 rows affected (0.00 sec)**

mysql> select user,host from mysql.user where user='root'; # 查询root用户授权情况

**+------+-----------+**

**| user | host |**

**+------+-----------+**

**| root | % |**

**| root | localhost |**

**+------+-----------+**

**2 rows in set (0.00 sec)**

mysql> exit; # 退出MySQL数据库

**Bye**

#### 步骤三：配置Hive组件

（1）设置Hive环境变量并使其生效。

# 在文件末尾追加以下配置内容

[root@master ~]# vi /etc/profile

# set hive environment

export HIVE\_HOME=/usr/local/src/hive

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

（2）修改Hive组件配置文件。

切换到hadoop用户执行以下对Hive组件的配置操作。

将/usr/local/src/hive/conf文件夹下hive-default.xml.template文件，更名为hive-site.xml。

[root@master ~]# su - hadoop

[hadoop@master ~]$ cp /usr/local/src/hive/conf/hive-default.xml.template /usr/local/src/hive/conf/hive-site.xml

（3）通过vi编辑器修改hive-site.xml文件实现Hive连接MySQL数据库，并设定Hive临时文件存储路径。

[hadoop@master ~]$ vi /usr/local/src/hive/conf/hive-site.xml

1）设置MySQL数据库连接。

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

<value>jdbc:mysql://master:3306/hive?createDatabaseIfNotExist=true&amp;useSSL=false</value>

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

2）配置MySQL数据库root的密码。

<property>

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

<value>Password123$</value>

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

</property>

3）配置数据库驱动。

<property>

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

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

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

</property>

4）配置数据库用户名javax.jdo.option.ConnectionUserName为root。

<property>

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

<value>root</value>

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

</property>

5）将以下位置的${system:java.io.tmpdir}/${system:user.name}替换为“/usr/local/src/hive/tmp”目录及其子目录。

需要替换以下4处配置内容：

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

<value>/usr/local/src/hive/tmp</value>

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

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

<value>/usr/local/src/hive/tmp</value>

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

<value>/usr/local/src/hive/tmp/resources</value>

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

<value>/usr/local/src/hive/tmp/operation\_logs</value>

6）在Hive安装目录中创建临时文件夹tmp。

[hadoop@master ~]$ mkdir /usr/local/src/hive/tmp

至此，Hive组件安装和配置完成。

#### 步骤四：初始化hive元数据

1）将MySQL数据库驱动（/opt/software/mysql-connector-java-5.1.46.jar）拷贝到Hive安装目录的lib下；

[hadoop@master ~]$ cp /opt/software/mysql-connector-java-5.1.46.jar /usr/local/src/hive/lib/

2）重新启动hadooop即可

[hadoop@master lib]$ stop-all.sh

[hadoop@master lib]$ start-all.sh

3）初始化数据库

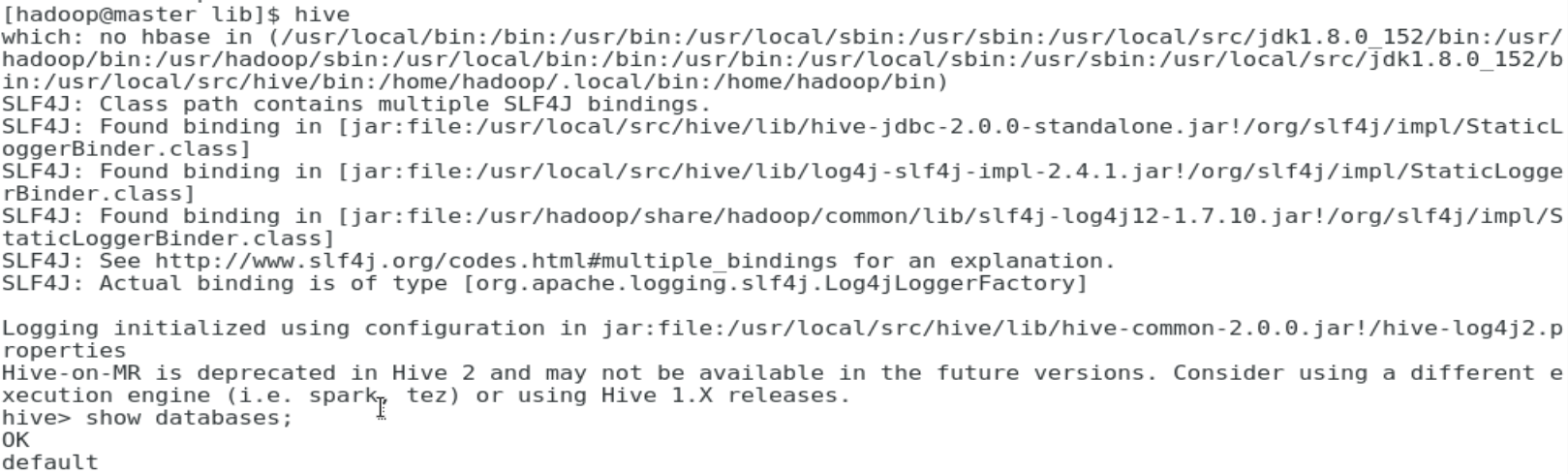
[hadoop@master ~]$schematool -initSchema -dbType mysql

4）启动hive

[hadoop@master ~]$ hive

注意: 出现namenode is in safe mode，再次执行hive即可。

这里hadoop启动后会自动关闭namenode的安全模式,关闭安全模式不会有任何提示,若执行完start-all.sh后立即执行hive,则会出现namenode is in safe mode的报错信息。



4）退出hive

[hadoop@master ~]$ exit;