目录

[1. 实验一：Sqoop组件安装与配置](#_Toc22000)

[1.1. 实验目的](#_Toc7576)

[1.2. 实验要求](#_Toc6632)

[1.3. 实验环境](#_Toc19919)

[1.4. 实验过程](#_Toc3978)

[1.4.1. 实验任务一：下载和解压Sqoop](#_Toc27192)

[1.4.2. 实验任务二：配置Sqoop环境](#_Toc4607)

[1.4.3. 实验任务三：启动Sqoop](#_Toc27875)

[1.4.4. 实验任务四：Sqoop模板命令](#_Toc14464)

[1.4.5. 实验任务五：Sqoop组件应用](#_Toc2047)

# 实验一：Sqoop组件安装与配置

## 实验目的

完成本实验，您应该能够：

* 下载和解压Sqoop
* 配置Sqoop环境
* 安装Sqoop
* Sqoop模板命令

## 实验要求

* 熟悉Sqoop环境
* 熟悉Sqoop模板命令

## 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Sqoop

Sqoop相关发行版本可以通过官网https://mirror-hk.koddos.net/apache/sqoop/来获取，如下图9-4所示。

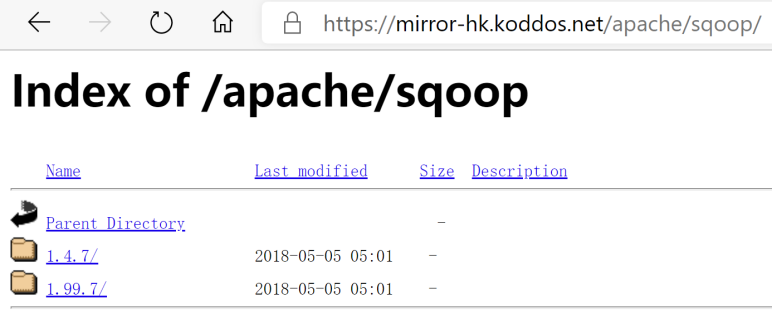


图9-4 Sqoop官网下载链接

安装Sqoop组件需要与Hadoop环境适配。使用root用户在Master节点上进行部署，将/opt/software/sqoop-1.4.7.bin\_\_hadoop-2.6.0.tar.gz压缩包解压到/usr/local/src目录下。

[root@master ~]# tar -zxvf /opt/software/sqoop-1.4.7.bin\_\_hadoop-2.6.0.tar.gz -C /usr/local/src

将解压后生成的sqoop-1.4.7.bin\_\_hadoop-2.6.0文件夹更名为sqoop。

[root@master ~]# cd /usr/local/src/

[root@master src]#mv ./sqoop-1.4.7.bin\_\_hadoop-2.6.0 sqoop

### 实验任务二：配置Sqoop环境

#### 步骤一：创建Sqoop的配置文件sqoop-env.sh。

复制sqoop-env-template.sh模板，并将模板重命名为sqoop-env.sh。

[root@master ~]# cd /usr/local/src/sqoop/conf/

[root@master conf]# cp sqoop-env-template.sh sqoop-env.sh

#### 步骤二：修改sqoop-env.sh文件，添加Hdoop、Hbase、Hive等组件的安装路径。

注意，下面各组件的安装路径需要与实际环境中的安装路径保持一致。

[root@master conf]# vi sqoop-env.sh

export HADOOP\_COMMON\_HOME=/usr/local/src/hadoop

export HADOOP\_MAPRED\_HOME=/usr/local/src/hadoop

export HBASE\_HOME=/usr/local/src/hbase

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

#### 步骤三：配置Linux系统环境变量，添加Sqoop组件的路径。

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

#在文件末尾添加

# set sqoop environment

export SQOOP\_HOME=/usr/local/src/sqoop

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

export CLASSPATH=$CLASSPATH:$SQOOP\_HOME/lib

#### 步骤四：连接数据库

为了使Sqoop能够连接MySQL数据库，需要将/opt/software/mysql-connector-java-5.1.46.jar文件放入sqoop的lib目录中。该jar文件的版本需要与MySQL数据库的版本相对应，否则Sqoop导入数据时会报错。（mysql-connector-java-5.1.46.jar对应的是MySQL 5.7版本）若该目录没有jar包,则使用第6章导入home目录的jar包

[root@master ~]# cp /opt/software/mysql-connector-java-5.1.46.jar /usr/local/src/sqoop/lib/

### 实验任务三：启动Sqoop

#### 步骤一：执行Sqoop前需要先启动Hadoop集群。

在master节点切换到hadoop用户执行start-all.sh命令启动Hadoop集群。

[root@master ~]# su - hadoop

[hadoop@master ~]$ source /etc/profile

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

#### 步骤二：检查Hadoop集群的运行状态。

[hadoop@master ~]$ jps

**1457 NameNode**

**1795 ResourceManager**

**2060 Jps**

**1646 SecondaryNameNode**

#### 步骤三：测试Sqoop是否能够正常连接MySQL数据库。

# Sqoop连接MySQL数据库 P大写 密码Password123$

[hadoop@master ~]$

sqoop list-databases --connect jdbc:mysql://127.0.0.1:3306/ --username root -P

**Warning: /home/hadoop/sqoop/../hcatalog does not exist! HCatalog jobs will fail.**

**Please set $HCAT\_HOME to the root of your HCatalog installation.**

**Warning: /home/hadoop/sqoop/../accumulo does not exist! Accumulo imports will fail.**

**Please set $ACCUMULO\_HOME to the root of your Accumulo installation.**

**Warning: /home/hadoop/sqoop/../zookeeper does not exist! Accumulo imports will fail.**

**Please set $ZOOKEEPER\_HOME to the root of your Zookeeper installation.**

**19/04/22 18:54:10 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7**

**Enter password: # 此处需要输入mysql数据库的密码**

**19/04/22 18:54:14 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.**

**information\_schema**

**hive**

**mysql**

**performance\_schema**

**sys**

能够查看到MySQL数据库中的information\_schema、hive、mysql、performance\_schema、sys等数据库，说明Sqoop可以正常连接MySQL。

#### 步骤四：连接hive

为了使Sqoop能够连接Hive，需要将hive组件/usr/local/src/hive/lib 目录下的hive-common-2.0.0.jar也放入Sqoop安装路径的lib目录中。

[hadoop@master ~] cp /usr/local/src/hive/lib/hive-common-2.0.0.jar /usr/local/src/sqoop/lib/

### 实验任务四：Sqoop模板命令

#### 步骤一：创建MySQL数据库和数据表。

创建sample数据库，在sample中创建student表，在student表中插入了3条数据。

# 登录MySQL数据库

[hadoop@master ~]$ mysql -uroot -p

**Enter password:**

# 创建sample库

mysql> create database sample;

**Query OK, 1 row affected (0.00 sec)**

# 使用sample库

mysql> use sample;

**Database changed**

mysql> create table student(number char(9) primary key, name varchar(10));

**Query OK, 0 rows affected (0.01 sec) # 创建student表，该数据表有number学号和name姓名两个字段**

# 向student表插入几条数据

mysql> insert into student values('01','zhangsan');

**Query OK, 1 row affected (0.05 sec)**

mysql> insert into student values('02','lisi');

**Query OK, 1 row affected (0.01 sec)**

mysql> insert into student values('03','wangwu');

**Query OK, 1 row affected (0.00 sec)**

mysql>

mysql>

# 查询student表的数据

mysql> select \* from student;

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

**| number | name |**

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

**| 01 | zhangsan |**

**| 02 | lisi |**

**| 03 | wangwu |**

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

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

mysql>

mysql> exit

#### 步骤二：在Hive中创建sample数据库和student数据表。

[hadoop@master ~]$ hive # 启动hive命令行

**Logging initialized using configuration in jar:file:/usr/local/src/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/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]**

**SLF4J: Found binding in [jar:file:/usr/local/src/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 explanation.**

**SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]**

hive> create database sample; # 创建sample库

**OK**

**Time taken: 0.679 seconds**

hive> show databases; # 查询所有数据库

**OK**

**default**

**Time taken: 0.178 seconds, Fetched: 2 row(s)**

hive> use sample; # 使用sample库

**OK**

hive> create table student(number STRING, name STRING)

**row format delimited**

**fields terminated by "|"**

**stored as textfile; # 创建student表**

**OK**

hive> exit; # 退出hive命令行

#### 步骤三：从MySQL导出数据，导入Hive。

需要说明该命令的以下几个参数：

1）--connect：MySQL数据库连接URL。

2）--username和--password：MySQL数据库的用户名和密码。

3）--table：导出的数据表名。

4）--fields-terminated-by：Hive中字段分隔符。

5）--delete-target-dir：删除导出目的目录。

6）--num-mappers：Hadoop执行Sqoop导入导出启动的map任务数。

7）--hive-import --hive-database：导出到Hive的数据库名。

8）--hive-table：导出到Hive的表名。

[hadoop@master ~]$ sqoop

import --connect jdbc:mysql://master:3306/sample --username root --password Password123$ --table student --fields-terminated-by '|' --delete-target-dir --num-mappers 1 --hive-import --hive-database sample --hive-table student

删除MySQL数据， student表中number为主键，添加信息导致主键重复，报错，所以删除表数据

[hadoop@master ~]$mysql -u root -p #密码Password123$

mysql>use sample;

mysql>delete from student;

mysql>exit;

#### 步骤四：从Hive导出数据，导入到MySQL。

需要说明该命令的以下几个参数：

1）--connect：MySQL数据库连接URL。

2）--username和--password：MySQL数据库的用户名和密码。

3）--table：导出的数据表名。

4）--fields-terminated-by：Hive中字段分隔符。

6）--export-dir：Hive数据表在HDFS中的存储路径。

[hadoop@master ~]$ sqoop export --connect "jdbc:mysql://master:3306/sample?useUnicode=true&characterEncoding=utf-8" --username root --password Password123$ --table student --input-fields-terminated-by '|' --export-dir /user/hive/warehouse/sample.db/student/\*

### 实验任务五：Sqoop组件应用

Sqoop常用设置命令如下：

（1）列出MySQL数据库中的所有数据库。

[hadoop@master ~]$ sqoop list-databases -connect jdbc:mysql://localhost:3306/ -username root -password Password123$

（2）连接MySQL并列出sample数据库中的表。

[hadoop@master ~]$ sqoop list-tables -connect jdbc:mysql://localhost:3306/sample -username root -password Password123$

（3）将关系型数据的表结构复制到hive中,只是复制表的结构，表中的内容没有复制过去。

[hadoop@master ~]$ sqoop create-hive-table -connect jdbc:mysql://localhost:3306/sample -table student -username root -password Password123$ -hive-table test

其中–table student为MySQL中的数据库sample中的表–hive-table test为hive中新建的表名称。

（4）从关系数据库导入文件到Hive中。

[hadoop@master ~]$sqoop import --connect jdbc:mysql://master:3306/sample --username root --password Password123$ --table student --delete-target-dir --num-mappers 1 --hive-import --hive-database default --hive-table test

1. 将Hive中的表数据导入到MySQL中,在进行导入之前，MySQL中的表

hive\_test表必须已经提前创建好。

删除MySQL数据， student表中number为主键，添加信息导致主键重复，报错，所以删除表数据

[hadoop@master ~]$mysql -u root -p #密码Password123$

mysql>use sample;

mysql>delete from student;

mysql>exit;

[hadoop@master ~]$ sqoop export -connect jdbc:mysql://master:3306/sample -username root -password Password123$ -table student --input-fields-terminated-by '\001' -export-dir /user/hive/warehouse/test

（6）从数据库导出表的数据到HDFS上文件。

[hadoop@master ~]$ sqoop import -connect jdbc:mysql://master:3306/sample -username root -password Password123$ -table student --num-mappers 1 -target-dir /user/test

（7）从数据库增量导入表数据到HDFS中。

#密码Password123$

[hadoop@master ~]$mysql -u root -p

mysql>use sample;

mysql> insert into student values('04','sss');

mysql> insert into student values('05','ss2');

mysql> insert into student values('06','ss3');

#非数值型的值不能当做增量

mysql> alter table student modify column number int;

mysql> exit;

[hadoop@master ~]$sqoop import -connect jdbc:mysql://master:3306/sample -username root -password Password123$ -table student --num-mappers 1 -target-dir /user/test -check-column number -incremental append -last-value 0

查看导入数据

[hadoop@master ~]$hdfs dfs -cat /user/test/part-m-00000