**UDH 之 Sqoop服务的安装和使用**

目录

[1. Sqoop简介 1](#_Toc430770074)

[2. 安装Sqoop 1](#_Toc430770075)

[3. 查看和配置Sqoop 2](#_Toc430770076)

[4. 运行服务检测 3](#_Toc430770077)

[5. Sqoop使用 4](#_Toc430770078)

[5.1. Mysql向HDFS导入数据 4](#_Toc430770079)

[5.2. HDFS向Mysql导入数据 4](#_Toc430770080)

[5.3. 创建Hive表 5](#_Toc430770081)

[5.4. 导入数据到Hive 5](#_Toc430770082)

# Sqoop简介

Sqoop是一款开源的工具，主要用于在Hadoop(Hive)与传统的数据库(mysql、postgresql...)间进行数据的传递，可以将一个关系型数据库（例如 ： MySQL ,Oracle ,Postgres等）中的数据导进到Hadoop的HDFS中，也可以将HDFS的数据导进到关系型数据库中。

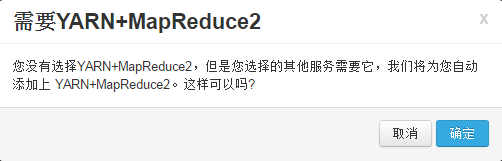
sqoop 主要通过 JDBC 和关系数据库进行交互。理论上支持 JDBC 的 database 都可以使用 sqoop 和 hdfs 进行数据交互。sqoop 架构非常简单，其整合了 Hive 、 Hbase 和 Oozie ，通过 map-reduce 任务来传输数据，从而提供并发特性和容错。

# 安装Sqoop

UDH安装向导第四步“选择服务”勾选Sqoop服务，一旦选择该服务，强制勾选HDFS， Zookeeper，Yarn等服务：



需要的服务未安装提示类似如下信息：



由于Sqoop仅有客户端，因此，不需要为其分配Master， Slaves 或者 Clients。

接着按照安装流程操作即可成功安装。

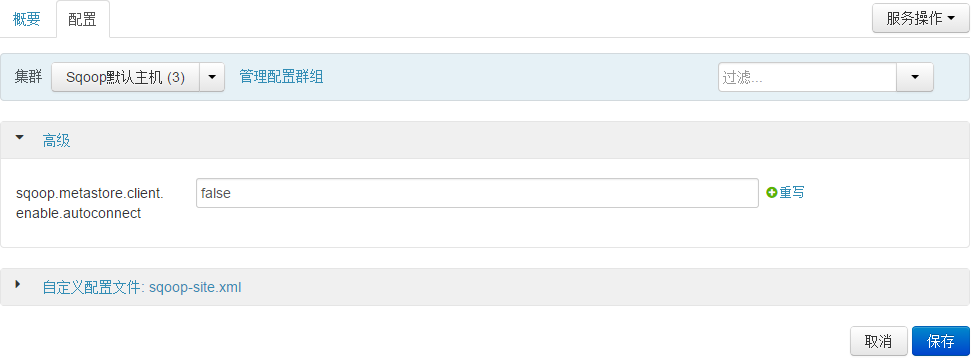
# 查看和配置Sqoop

安装完成后，点击“Sqoop”进入Sqoop服务界面：



Sqoop界面仅有概要信息，显示Sqoop客户端已安装。

点击配置页签查看Sqoop服务的相关配置参数：



配置参数的修改时，直接填写对应的修改项，点击保存，选择重新启动按钮即可令修改生效。

# 运行服务检测

点击Sqoop“服务操作”下的“运行服务检测”可验证服务是否能够正常运行：



运行服务检测成功说明Sqoop服务成功启用：



# Sqoop使用

## Mysql向HDFS导入数据

Sqoop只能通过命令行进行操作本文举例从MySql导入数据到Hdfs，测试 sqoop 连接 mysql：

# /usr/lib/sqoop/bin/sqoop-list-databases --connect jdbc:mysql://20.12.20.12:3306/ --username root

Warning: /usr/lib/sqoop/bin/../../accumulo does not exist! Accumulo imports will fail.

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

15/08/27 16:06:35 INFO sqoop.Sqoop: Running Sqoop version: 1.4.5-cdh5.4.3

15/08/27 16:06:35 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.

information\_schema

mysql

sqoop

test

看到了mysql , test 库，则说明连接没问题。

使用 sqoop import 命令可以从关系数据库导入数据到 hdfs，具体实例如下：

sqoop import --connect jdbc:mysql://20.12.20.11:3306/hive --username root --password 123456 --table TBLS --columns "tbl\_id,create\_time" --where "tbl\_id > 1" --target-dir /user/hive/result

查看 hdfs 中的结果数据：

# hadoop fs -cat /user/hive/result/part-m-00000

2,1440639469

## HDFS向Mysql导入数据

查看hdfs 中的测试数据：

# hadoop fs -cat /user/root/test.txt

1,aaaaaa,bbbb

2,cccccc,dddd

3,eeeeee,ffff

执行下面的命令会将 hdfs 中的数据导入到 mysql 中：

#sqoop export --connect jdbc:mysql://20.12.20.12:3306/sqoop --username root --table yonyou --export-dir '/user/root/test.txt' --fields-terminated-by ','

查看结果

mysql> select \* from yonyou;

+------+--------+------+

| id   | va01   | va02 |

+------+--------+------+

|    1 | aaaaaa | bbbb |

|    2 | cccccc | dddd |

|    3 | eeeeee | ffff |

+------+--------+------+

3 rows in set (0.00 sec)

## 创建Hive表

生成与关系数据库表的表结构对应的 HIVE 表：

#sqoop create-hive-table --connect jdbc:mysql://20.12.20.12:3306/sqoop  --username root --table yonyou

note ：

--hive-home <dir>   Hive 的安装目录，可以通过该参数覆盖掉默认的 hive 目录

--hive-overwrite    覆盖掉在 hive 表中已经存在的数据

--create-hive-table 默认是 false ，如果目标表已经存在了，那么创建任务会失败

--hive-table        后面接要创建的 hive 表

--table             指定关系数据库表名

## 导入数据到Hive

执行下面的命令会将 mysql 中的数据导入到 hdfs 中，然后创建一个 hive 表，最后再将 hdfs 上的文件移动到 hive 表的目录下面。

#sqoop import --connect jdbc:mysql://20.12.20.11:3306/hive --username root --password 123456 --table TBLS --fields-terminated-by "\t" --lines-terminated-by "\n" --hive-import --hive-overwrite --create-hive-table --hive-table dw\_srclog.TBLS --delete-target-dir

注意:

* 可以在 hive 的表名前面指定数据库名称
* 可以通过 --create-hive-table 创建表，如果表已经存在则会执行失败

接下来可以查看 hive 中的数据：

# hive -e 'select \* from dw\_srclog.tbls'

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties

OK

2       1440639469      1       0       hue     0       2       test    MANAGED\_TABLE   null    null    NULL

3       1440664128      1       0       hue     0       3       test\_yonyou     MANAGED\_TABLE   null    null    NULL

4       1440664159      1       0       hue     0       4       test\_you        MANAGED\_TABLE   null    null    NULL

Time taken: 3.02 seconds, Fetched: 3 row(s)

直接查看文件内容：

# hadoop fs -cat /user/hive/warehouse/dw\_srclog.db/tbls/part-m-0000\*

2       1440639469      1       0       hue     0       2       test    MANAGED\_TABLE   null    null    null

3       1440664128      1       0       hue     0       3       test\_yonyou     MANAGED\_TABLE   null    null    null

4       1440664159      1       0       hue     0       4       test\_you        MANAGED\_TABLE   null    null    null