

Ubuntu上安装与运行Sqoop

(1) Sqoop压缩包上传到服务器

略

(2) 解压Sqoop安装包

```
1 $ sudo tar -zxvf sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz -C /usr/local
```

(3) 重命名解压后的目录并修改目录所有者

```
1 $ cd /usr/local
2 $ sudo mv sqoop-1.4.7.bin__hadoop-2.6.0-alpha sqoop
3 $ sudo chown -R hadoop:hadoop sqoop
```

(4) 配置环境变量。

```
1 $ gedit ~/.bashrc
```

在配置文件最后增加如下信息

```
1 export SQOOP_HOME=/usr/local/sqoop
2 export PATH=$PATH:$SQOOP_HOME/bin
3
4 export HADOOP_CLASSPATH=$HADOOP_HOME/lib/*
5 export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$HIVE_HOME/lib/*
6
7 export HIVE_CONF_DIR=$HIVE_HOME/conf
```

```
1 $ source ~/.bashrc
```

(5) 创建、编辑配置文件sqoop-env.sh

```
1 $ cd /usr/local/sqoop/conf
2 $ cp sqoop-env-template.sh sqoop-env.sh
3 $ gedit sqoop-env.sh
```

在sqoop-env.sh中写入下列信息：

```
1 export HADOOP_COMMON_HOME=/usr/local/hadoop
2 export HADOOP_MAPRED_HOME=/usr/local/hadoop
3 export HBASE_HOME=/usr/local/hbase
4 export HIVE_HOME=/usr/local/hive
5
```

(6) 将MySQL驱动包复制到\$SQOOP_HOME/lib目录

```
1 cd /mysql驱动的下载目录
2 cp ./mysql-connector-j-8.0.33.jar /usr/local/sqoop/lib
```

(7) 设置MySQL允许加载文件数据并测试与MySQL的连接

修改 /etc/mysql/mysql.conf.d/mysqld.conf, 在[mysqld]下面添加secure-file-priv = ""

```
1 [mysqld]
2 secure-file-priv = ""
```

重启mysql

```
1 sudo service mysql restart
```

连接mysql的时候，增加参数

```
1 mysql -uroot -p --local-infile=1
```

登录mysql之后，执行

```
1 SET GLOBAL local_infile = 'ON';
```

退出mysql

```
1 $ exit
```

测试Sqoop与MySQL之间的连接是否成功：

```
1 $ sqoop list-databases --connect jdbc:mysql://master:3306/ --username root --password root
2 $ sqoop list-tables --connect jdbc:mysql://master:3306/hive --username root --password root
```

(8) MySQL数据准备

连接mysql，注意增加--local-infile=1参数

```
1 $mysql -uroot -p --local-infile=1
```

```

1 mysql> CREATE DATABASE IF NOT EXISTS test;
2 mysql> USE test;
3 mysql> CREATE TABLE emp(empno int ,ename char(20),job char(20),mgr int,hiredate
  datetime,sal float,comm float,deptno int);
4 mysql>LOAD DATA LOCAL INFILE '/home/hadoop/data/emp.txt' INTO TABLE emp FIELDS
  TERMINATED BY ',';

```

(9) 数据从MySQL导入HDFS,全量导入

```

1 sqoop import --connect jdbc:mysql://localhost:3306/test -username root -password root -
  -table emp --columns empno,ename,hirrrdate,sal --where 'sal>5000' --target-dir
  /emps/emp2 --delete-target-dir --fileds-teminated by '\t' --m 2 --split-by empno

```

```

1 sqoop import --connect jdbc:mysql://localhost:3306/test --username root --password
  root --query 'select deptno,count(*) num,avg(sal) avgsal from emp WHERE $CONDITIONS
  GROUP BY deptno' --target-dir /emps/emp3 --delete-target-dir -m 1

```

指定query或者e参数传入一个SQL语句

```

1 sqoop import --connect jdbc:mysql://mashuai01:3306/test --username root --password
  root --query 'select * from emp where sal>2000 and job ="st_clerk" and $CONDITIONS' --
  target-dir /emps/emp4 --delete-target-dir -m 1

```

增量导入验证

先做一次全量导入

```

1 sqoop import --connect jdbc:mysql://mashuai01:3306/test --username root --password
  root --table emp --target-dir /emps/emp5 --delete-target-dir --m 2 --split-by empno
2

```

模拟数据库增加数据, 登录mysql并执行如下sql

```
1 INSERT INTO emp VALUES(207,'ZHANGSHAN','SH_CLERK',110,'2020-10-1',5000,500,20);
2 INSERT INTO emp VALUES(208,'LISI','SH_CLERK',110,'2020-5-10',3500,200,30);
```

增量导入

```
1 sqoop import --connect jdbc:mysql://mashuai01:3306/test --username root --password
  root --table emp --incremental append --check-column empno --last-value 206 --target-
  dir /emps/emp5 --m 2 --split-by empno
2
```

验证增量导入是否成功

```
1 hdfs dfs -cat /emps/emp5/part-m-00000
2 hdfs dfs -ls /emps/emp5
```

sqoop将mysql数据导入到Hive

```
1 sqoop import --connect jdbc:mysql://mashuai01:3306/test --username root --password
  root --table emp --hive-import --hive-database empdb --hive-table hiveemp -m 1
2
```

(10) 执行中可能遇到的问题

MySQL使用LOAD FILE数据无法导入的问题

```
1 ERROR 3948 (42000): Loading local data is disabled; this must be enabled on both the
  client and server sides
```

以上问题执行第7步可以解决

如果执行过程中遇到如下问题

```
1 2024-04-11 16:45:29,913 INFO hive.HiveImport: Loading uploaded data into Hive
2 2024-04-11 16:45:29,918 ERROR hive.HiveConfig: Could not load
  org.apache.hadoop.hive.conf.HiveConf. Make sure HIVE_CONF_DIR is set correctly.
3 2024-04-11 16:45:29,920 ERROR tool.ImportTool: Import failed: java.io.IOException:
  java.lang.ClassNotFoundException: org.apache.hadoop.hive.conf.HiveConf
4         at org.apache.sqoop.hive.HiveConfig.getHiveConf(HiveConfig.java:50)
5         at org.apache.sqoop.hive.HiveImport.getHiveArgs(HiveImport.java:392)
6         at
  org.apache.sqoop.hive.HiveImport.executeExternalHiveScript(HiveImport.java:379)
7
```

修改环境配置，将hive的lib添加到HADOOP_CLASSPATH中

```
1 vim ~/.bashrc
```

添加如下内容

```
1 export HADOOP_CLASSPATH=$HADOOP_HOME/lib/*
2 export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$HIVE_HOME/lib/*
```

```
1 source ~/.bashrc
```

为避免影响其他程序

可以

```
1 export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$HIVE_HOME/lib/*
2 # 运行 Sqoop 命令
3 ...
4 # 恢复原始环境变量
5 export HADOOP_CLASSPATH=${HADOOP_CLASSPATH%:$HIVE_HOME/lib/*}
```

这个命令的原理是使用 shell 中的参数替换特性来删除 `HADOOP_CLASSPATH` 中结尾的 `:$HIVE_HOME/lib/*` 部分。让我们逐步解释这个命令：

1. `${HADOOP_CLASSPATH%:$HIVE_HOME/lib/*}`：这是一种参数替换形式，其中 `${parameter%word}` 表示从变量 `parameter` 的末尾删除匹配的模式 `word`。在这里，`parameter` 是 `HADOOP_CLASSPATH`，`word` 是 `:$HIVE_HOME/lib/*`。这个模式表示结尾是以冒号、`$HIVE_HOME/lib/`、任意字符（`*`通配符表示任意字符）的字符串。
2. `%` 表示删除匹配模式的最小部分，而 `%%` 则表示删除匹配模式的最大部分。
3. `${HADOOP_CLASSPATH%:$HIVE_HOME/lib/*}` 将会从 `HADOOP_CLASSPATH` 的末尾删除与模式 `:$HIVE_HOME/lib/*` 匹配的部分。

通过这个命令，你可以移除 `HADOOP_CLASSPATH` 中最后面由 `$HIVE_HOME/lib/` 开头的部分，实现了环境变量的恢复。

如果在运行sqoop时

```
1 sqoop import --connect jdbc:mysql://localhost:3306/test --username root --password
  root --query 'select deptno,count(*) num,avg(sal) avgsal from emp WHERE $CONDITIONS
  GROUP BY deptno' --target-dir /emps/emp3 --delete-target-dir -m 1
```

出现如下错误：

```
1 Error: A JNI error has occurred, please check your installation and try again
2 Exception in thread "main" java.lang.NoClassDefFoundError:
  org.apache.hadoop.yarn.exceptions.YarnRuntimeException
3     at java.lang.Class.getDeclaredMethods0(Native Method)
4     at java.lang.Class.privateGetDeclaredMethods(Class.java:2701)
5     at java.lang.Class.privateGetMethodRecursive(Class.java:3048)
6     at java.lang.Class.getMethod0(Class.java:3018)
7     at java.lang.Class.getMethod(Class.java:1784)
8     at sun.launcher.LauncherHelper.validateMainClass(LauncherHelper.java:544)
9     at sun.launcher.LauncherHelper.checkAndLoadMain(LauncherHelper.java:526)
10 Caused by: java.lang.ClassNotFoundException:
  org.apache.hadoop.yarn.exceptions.YarnRuntimeException
11     at java.net.URLClassLoader.findClass(URLClassLoader.java:381)
12     at java.lang.ClassLoader.loadClass(ClassLoader.java:424)
13     at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:349)
14
```