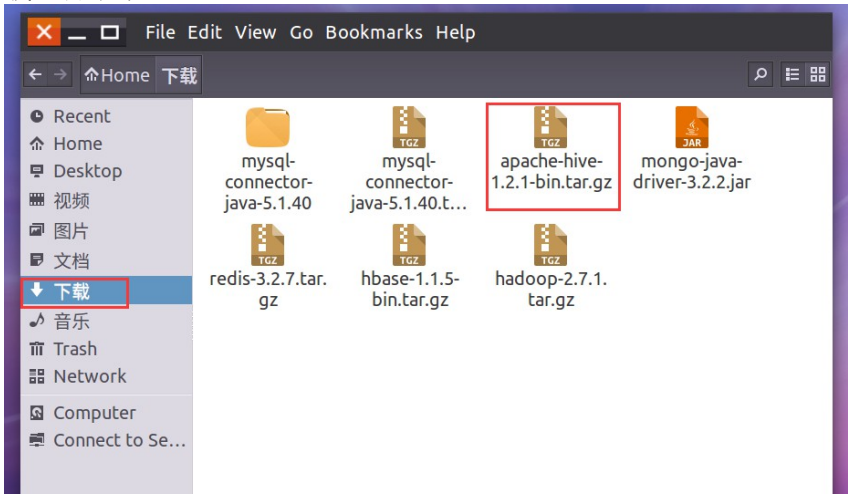
第 6 篇：数据仓库 Hive 的安装和使用

学习 blog：https://blog.csdn.net/achuo/article/details/51332214

一、Hive 的安装

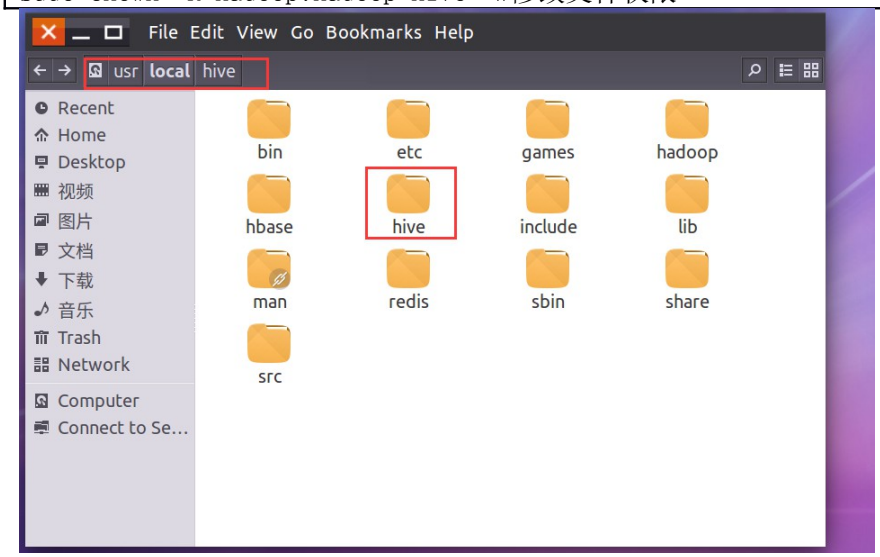
1. 下载安装文件，解压、并设置权限

（1）在本书软件下载专区下载文件“apache-hive-1.2.1-bin.tar.gz”，放入"/home/下载”目录下。



（2）解压文件“apache-hive-1.2.1-bin.tar.gz”到"/usr/local/"目录下，并重命名和修改权限

```
sudo tar -zxvf ./下载/apache-hive-1.2.1-bin.tar.gz -C /usr/local  #解压
cd /usr/local/
sudo mv apache-hive-1.2.1-bin hive  #重命名
sudo chown -R hadoop:hadoop hive  #修改文件权限
```



2. 配置环境变量

打开配置文件

```
vim ~/.bashrc
```

在文件 PATH 目录最前面添加如下内容：

```
export PATH=/usr/local/hive/bin:
```

使配置生效

```
source ~/.bashrc
```

3. 修改配置文件

(1)将/usr/local/hive/conf 目录下的 hive-default.xml.template 文件重命名为 hive-default.xml，命令如下：

```
cd /usr/local/hive/conf
sudo mv hive-default.xml.template hive-default.xml
```

(2)新建 hive-site.xml 文件，输入配置信息。

命令如下：

```
cd /usr/local/hive/conf
vim hive-site.xml
```

hive-site.xml 中的内容如下：

（注意：第 1 句和书上不同！！）

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>javax.jdo.option.ConnectionURL</name>
                                        <value>jdbc:mysql://localhost: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>hive</value>
    <description>username to use against metastore database</description>
  </property>
  <property>
    <name>javax.jdo.option.ConnectionPassword</name>
    <value>hive</value>
    <description>password to use against metastore database</description>
  </property>
</configuration>
```

4. 安装并配置 MySQL

(1)安装 MySQL

执行安装命令：

```
sudo apt-get update
sudo apt-get install mysql-server
```

启动 MySQL 服务，命令如下：

```
service mysql stop
service mysql start
```

确认是否服务是否启动成功，命令如下：

```
sudo netstat -tap | grep mysql
```

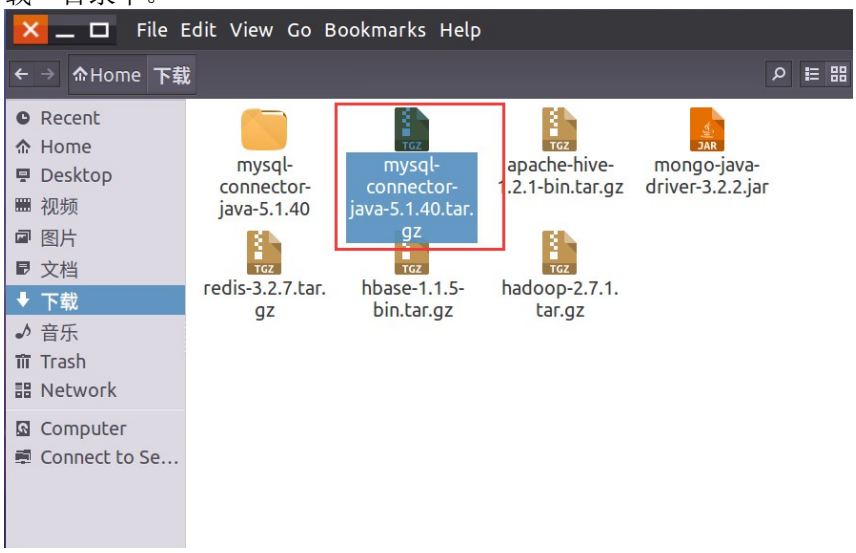
若 MySQL 节点处于“LISTEN”状态，则表示启动成功。

（2）无密码问题请参看技术文档：“技术文档（1）：MySQL 安装无密码解决办法”

解决后，root 密码为“123”

（3）下载 MySQL JDBC 驱动程序

在本书软件下载专区下载文件“mysql-connector-java-5.1.40.tar.gz”，放入"/home/下载”目录下。



解压成 jar 包并复制到“/usr/local/hive/lib”目录下

```
cd ~/下载
tar -zxvf mysql-connector-java-5.1.40.tar.gz  #解压
cd ./mysql-connector-java-5.1.40
cp mysql-connector-java-5.1.40-bin.jar /usr/local/hive/lib    #复制
```

（4）启动 MySQL

```
service mysql start
mysql -u root -p
输入密码：1
```

（5）在 MySQL 中为 Hive 新建数据库

在“mysql>”命令提示符下，输入命令：

```
create database hive;
```

（6）配置 MySQL 允许 Hive 接入

对 MySQL 进行权限配置，允许 Hive 连接到 MySQL。在“mysql>”命令提示符下，输入命令：

```
grant all on *.* to hive@localhost identified by 'hive';  #将所有数据库所有表的所有权限赋给 hive 用户
flush privileges;    #刷新权限
```

（7）启动 Hive：先启动 hadoop，再启动 Hive

```
cd /usr/local/hadoop
./sbin/start-dfs.sh    #启动 hadoop
cd /usr/local/hive
./bin/hive
```

（8）错误解决

若启动 Hive 时，出现“Hive metastore database is not initialized”的错误。因为曾经安装了 Hive 或 MySQL，重新安装 Hive 和 MySQL 以后，导致版本和配置不一致。解决方法是使用 schematool 工具。此工具用于初始化当前 Hive 版本的 Metastore 架构。在终端下输入如下命令：

```
schematool -dbType mysql -initSchema
```

该命令执行后，再启动 Hive，就可以正常启动了。

二、Hive 的数据类型

1.Hive 基本数据类型

整型、浮点型、布尔型、无长度限制的字符串型、时间戳类型、二进制数组类型
INT、DOUBLE、BOOLEAN、STRING、TIMESTAMP、BINARY

2.Hive 集合数据类型

struct：一组命名的字段，字段类型可以不同。 Struct('a',1,1,0)
map：一组无序的键/值对，键的类型必须是原子的，值可以是任何数据类型，同一个映射的键和值的类型必须相同。 Map('a',1,'b',2)
array：一组有序字段，字段的类型必须相同。 Array(1,2)

三、Hive 基本操作

1.创建数据库

```
hive>create database if not exists hive;
```

2.创建表

（1）创建表，含 3 个属性

```
hive>use hive;
hive>create table if not exists usr(id bigint,name string,age int);
```

（2）创建表的同时设定存储路径

```
hive> create table if not exists usr(id bigint,name string,age int)
   > location '/usr/local/hive/warehouse/hive/usr';
```

（3）在 hive 数据库中创建外部表 usr，可以读取路径“/usr/local/data”下以“,”分割的数据

```
hive>create external table if not exists hive.usr(id bigint,name string,age int)
   >row format delimited fields terminated by ','
   >location '/usr/local/data'
```

（4）在 hive 数据库中创建分区表 usr,还存在分区字段 sex

```
hive>create table hive.usr(id bigint,name string,age int) partition by(sex boolean);
```

（5）在 hive 数据库中创建分区表 usr1,它通过复制表 usr 得到

```
hive>use hive
hive>create table if not exists usr1 like usr;
```

3.创建视图

创建视图 little_usr，只包含 usr 表中的 id、age 属性

```
hive>create view little_usr as select id,age from usr;
```

4.删除数据库

(1)删除数据库 hive

```
hive>drop database if exists hive;
```

(2)删除数据库和数据库中的表

```
hive>drop database if exists hive cascade;
```

5.删除表

删除 usr，如果是内部表，元数据和实际数据都会被删除；如果是外部表，只删除元数据，不删除实际数据。

```
hive>drop table if exists usr;
```

6.删除视图

```
hive>drop view if exists little_usr;
```

7.修改数据库
为hive数据库设置dbproperties键值对属性值来描述数据库属性信息。
```
hive>alter database hive set dbproperties('edited-by'='lily');
```
8.修改表
（1）重命名表
```
hive>alter table usr rename user;
```
（2）为表usr增加新分区
```
hive>alter table usr add if not exists partition(age=10);
```
(3) 删除表usr中的分区
```
hive>alter table usr drop if exists partition(age=10);
```
(4)修改表的列名：name修改为usename，并把该列置于age列后。
```
hive>alter table usr change name username string after age;
```
（5）增加一个新列
```
hive>alter table usr add columns(sex boolean);
```
（6）删除表的字段并重新指定字段
```
hive>alter table usr replace columns(newid bigint,newname string,newage int);
```
（7）为表usr设置tblproperties键值对属性值来描述表的属性信息。
```
hive>alter table usr set tblproperties('notes'='the columns in usr may be null except id');
```
9.修改视图
```
hive>alter view little_usr set tblproperties('create_at'='refer to timestamp');
```
10.查看数据库、表和视图
（1）查看数据库
```
hive>show databases;
hive>show databases like 'h.*';  #查看以h开头的所有数据库
```
（2）查看表和视图
```
hive>use hive;
hive>show tables;  #查看hive中的所有表和视图
hive>show tables in hive like 'u.*';  #查看数据库中以u开头的所有表和视图
```
11.描述数据库、表和视图
```
hive>describe database hive;    #查看数据库基本信息
hive>describe database extended hive;   #查看数据库详细信息
hive>describe hive.usr;        #查看表基本信息
hive>describe hive.little_usr;   #查看视图基本信息
hive>describe extended hive.usr;   #查看表usr的详细信息
hive>describe extended hive.little_usr;  #查看视图little_usr的详细信息
hive>describe extended hive.usr.id;    #查看表usr中列id的信息
```
12.向表中装载数据
（1）把目录“/usr/local/data”下的数据文件中的数据装载进usr表并覆盖原有数据。
```
hive>load data local inpath '/usr/local/data' overwrite into table usr;
```
（2）把目录“/usr/local/data”下的数据文件中的数据装载进usr表，不覆盖原有数据。
```
hive>load data local inpath '/usr/local/data' into table usr;
```
（3）装载分布式文件系统目录“hdfs://master_server/usr/local/data”下的数据文件数据装载进usr表并覆盖原有数据。
```
hive>load data inpath ' hdfs://master_server/usr/local/data ' overwrite into table usr;
```
13.向表中插入数据或从表中导出数据
（1）向表usr1中插入来着usr表的数据并覆盖原有数据
```
hive>insert overwrite table usr1
   >select * from usr where age=10;
```

（2）向表 usr1 中插入来着 usr 表的数据并追加在原有数据后

```
hive>insert into table usr1
   >select * from usr where age=10;
```

四、Hive 应用实例

1. 单机模式下运行实例

（1）创建 input 目录

```
cd /usr/local/hadoop
mkdir input
```

（2）创建测试文档

```
cd input
echo "hello world">file1.txt
echo "hello hadoop">file2.txt
```

（3）进入 hive 命令行，编写 HiveQL 语句实现 WordCount 算法，命令如下：

```
hive>create table docs(line string);
hive>load data local inpath '/usr/local/hadoop/input' overwrite into table
docs;
hive>create table word_count as
     select word, count(1) as count from
     (select explode(split(line,' '))as word from docs) w
     group by word
     order by word;
hive>select * from word_count;
```

```
hive> select * from word_count
    > ;
OK
hadoop  1
hello   2
world   1
```

2.伪分布模式下运行实例

（1）在伪分布模式下创建文件夹 input1，将文件 wordfile1.txt 复制到 input1 中。

```
cd /usr/local/hadoop
sbin/start-dfs.sh
bin/hdfs dfs -mkdir -p  /user/hadoop  #在 HDFS 中创建用户目录
bin/hdfs dfs -mkdir input1      #在 HDFS 中创建 hadoop 用户对于的 input1 目录
bin/hdfs dfs -ls input1   #查看 HDFS 中 input1 目录下的文件
bin/hdfs dfs -put ~/wordfile1.txt input1  #拷贝文件 wordfile1.txt 到 input1 中
bin/hdfs dfs -ls input1  #查看
```

```
hadoop@ubuntu:~$ cd /usr/local/hadoop
hadoop@ubuntu:/usr/local/hadoop$ sbin/start-dfs.sh
Starting namenodes on [localhost]
localhost: namenode running as process 3940. Stop it first.
localhost: datanode running as process 4065. Stop it first.
Starting secondary namenodes [0.0.0.0]
0.0.0.0: secondarynamenode running as process 4282. Stop it first.
hadoop@ubuntu:/usr/local/hadoop$ ./bin/hdfs dfs - mkdir -p /user/hadoop
mkdir: Unknown command
Did you mean -mkdir?  This command begins with a dash.
hadoop@ubuntu:/usr/local/hadoop$ ./bin/hdfs dfs -mkdir -p /user/hadoop
hadoop@ubuntu:/usr/local/hadoop$ ./bin/hdfs dfs -mkdir input1
mkdir: `input1': File exists
hadoop@ubuntu:/usr/local/hadoop$ ./bin/hdfs dfs -ls input1
hadoop@ubuntu:/usr/local/hadoop$ ./bin/hdfs dfs -put ~/wordfile1.txt input1
hadoop@ubuntu:/usr/local/hadoop$ ./bin/hdfs dfs -ls input1
Found 1 items
-rw-r--r--   1 hadoop supergroup         28 2018-08-19 19:32 input1/wordfile1.tx
t
```

（2）进入 hive 命令行，编写 HiveQL 语句实现 WordCount 算法，命令如下：

```
hive>create table if not exists docs(line string);
hive>load data inpath 'input1' overwrite into table docs;
hive>drop table word_count;
hive>create table word_count as
     select word, count(1) as count from
     (select explode(split(line,' '))as word from docs) w
     group by word
     order by word;
hive>select * from word_count;
```

```
hive> select * from word_count;
OK
        1
Hadoop  1
I       2
Spark   1
love    2
```

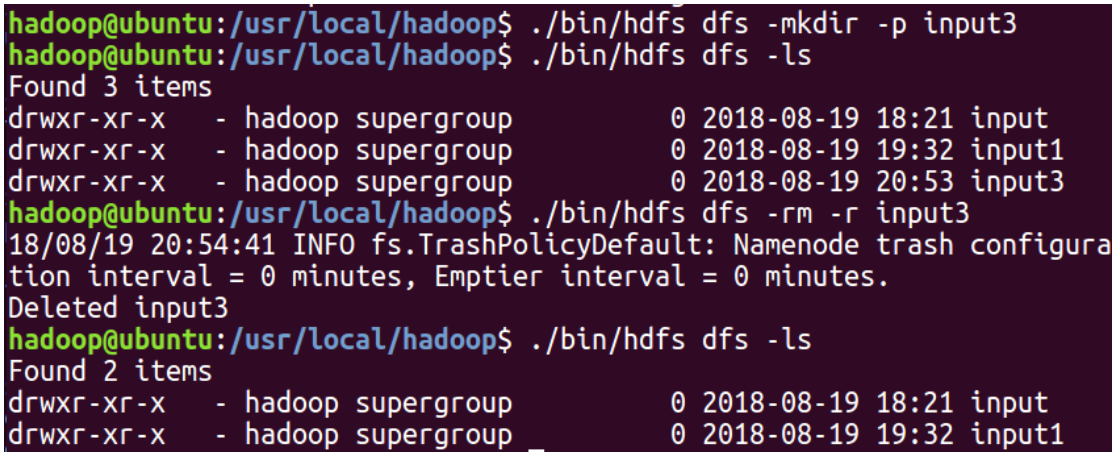
附件：

HDFS 文件操作：

1.创建文件夹：./bin/hdfs dfs -mkdir -p input3

2.查看所有文件夹：./bin/hdfs dfs -ls

3.删除文件夹：./bin/hdfs dfs -rm -r input3


hadoop@ubuntu:/usr/local/hadoop$ ./bin/hdfs dfs -mkdir -p input3
hadoop@ubuntu:/usr/local/hadoop$ ./bin/hdfs dfs -ls
Found 3 items
drwxr-xr-x - hadoop supergroup 0 2018-08-19 18:21 input
drwxr-xr-x - hadoop supergroup 0 2018-08-19 19:32 input1
drwxr-xr-x - hadoop supergroup 0 2018-08-19 20:53 input3
hadoop@ubuntu:/usr/local/hadoop$ ./bin/hdfs dfs -rm -r input3
18/08/19 20:54:41 INFO fs.TrashPolicyDefault: Namenode trash configura
tion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted input3
hadoop@ubuntu:/usr/local/hadoop$ ./bin/hdfs dfs -ls
Found 2 items
drwxr-xr-x - hadoop supergroup 0 2018-08-19 18:21 input
drwxr-xr-x - hadoop supergroup 0 2018-08-19 19:32 input1