1. 下载hive

cd /usr/local

wget http://mirror.bit.edu.cn/apache/hive/hive-2.3.2/[apache-hive-2.3.2-bin.tar.gz](http://mirror.bit.edu.cn/apache/hive/hive-2.3.2/apache-hive-2.3.2-bin.tar.gz)

tar -zxvf apache-hive-2.3.2-bin.tar.gz

1. 配置环境变量

vi /etc/profile

export HIVE\_HOME=/usr/local/apache-hive-2.3.2-bin

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

source /etc/profile

1. hive 配置 hdfs

cd /usr/local/apache-hive-2.3.2-bin/conf

cp hive-default.xml.template hive-site.xml

新建 hdfs 目录

使用 hadoop 新建 hdfs 目录,因为在 hive-site.xml 中有默认如下配置：

<property>

<name>hive.metastore.warehouse.dir</name>

<value>/opt/hive/warehouse</value>

<description>location of default database for the warehouse</description>

</property>

<property>

进入 hadoop 安装目录 执行hadoop命令新建/user/hive/warehouse目录，并授权，用于存储文件

cd /home/hadoop/hadoop-2.7.4

bin/hadoop fs -mkdir -p /opt/hive/warehouse

bin/hadoop fs -mkdir -p /opt/hive/tmp

bin/hadoop fs -mkdir -p /opt/hive/log

bin/hadoop fs -chmod -R 777 /opt/hive/warehouse

bin/hadoop fs -chmod -R 777 /opt/hive/tmp

bin/hadoop fs -chmod -R 777 /opt/hive/log

用以下命令检查目录是否创建成功

**bin/hadoop**fs -ls /opt/hive

修改 hive-site.xml

搜索hive.exec.scratchdir,将该name对应的value修改为/user/hive/tmp

<property>

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

<value>/opt/hive/tmp</value>

</property>

搜索hive.querylog.location,将该name对应的value修改为/user/hive/log/hadoop

<property>

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

<value>/opt/hive/log/hadoop</value>

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

</property>

搜索javax.jdo.option.connectionURL,将该name对应的value修改为MySQL的地址

<property>

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

<value>jdbc:mysql://192.168.0.70:3306/hive?createDatabaseIfNotExist=true</value>

<description>

JDBC connect string for a JDBC metastore.

To use SSL to encrypt/authenticate the connection, provide database-specific SSL flag in the connection URL.

For example, jdbc:postgresql://myhost/db?ssl=true for postgres database.

</description>

</property>

搜索javax.jdo.option.ConnectionDriverName，将该name对应的value修改为MySQL驱动类路径

<property>

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

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

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

</property>

搜索javax.jdo.option.ConnectionUserName，将对应的value修改为MySQL数据库登录名

<property>

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

<value>root</value>

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

</property>

搜索javax.jdo.option.ConnectionPassword，将对应的value修改为MySQL数据库的登录密码

<property>

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

<value>AAaa\_123456</value>

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

</property>

创建 tmp 文件

mkdir /home/hadoop/hive-2.3.0/tmp

并在 hive-site.xml 中修改

把{system:java.io.tmpdir} 改成 /home/hadoop/hive-2.3.0/tmp

把 {system:user.name} 改成 {user.name}

### 新建 hive-env.sh

cd conf

cp hive-env.sh.template hive-env.sh

vi hive-env.sh

### # Set HADOOP\_HOME to point to a specific hadoop install directory

### HADOOP\_HOME=/opt/hadoop-2.7.3/

### # Hive Configuration Directory can be controlled by:

### export HIVE\_CONF\_DIR=/usr/local/apache-hive-2.3.2-bin/conf

### # Folder containing extra libraries required for hive compilation/execution can be controlled by:

### export HIVE\_AUX\_JARS\_PATH=/usr/local/apache-hive-2.3.2-bin/lib

### 下载 mysql 驱动包

cd /usr/local/apache-hive-2.3.2-bin/lib

wget http://central.maven.org/maven2/mysql/mysql-connector-java/5.1.38/mysql-connector-java-5.1.38.jar

## 初始化 mysql

### MySQL数据库进行初始化

首先确保 mysql 中已经创建 hive 库

cd /usr/local/apache-hive-2.3.2-bin/bin

./schematool -initSchema -dbType mysql

如果看到如下,表示初始化成功

**Starting** **metastore** **schema** **initialization** **to** 2.3.0

**Initialization** **script** **hive-schema-2**.3.0.mysql.sql

**Initialization** **script** **completed**

**schemaTool** **completed**

### 查看 mysql 数据库

/usr/local/mysql/bin/mysql -uroot -p

**mysql> show databases;**

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

**| Database |**

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

| information*\_schema |*

*| hive |*

*| mysql |*

*| performance\_*schema |

**| sys |**

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

5 rows in set (0.00 sec)

mysql> use hive;

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

mysql> show tables;

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

| Tables\_in\_hive |

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

| AUX\_TABLE |

| BUCKETING\_COLS |

| CDS |

| COLUMNS\_V2 |

| COMPACTION\_QUEUE |

| COMPLETED\_COMPACTIONS |

| COMPLETED\_TXN\_COMPONENTS |

| DATABASE\_PARAMS |

| DBS |

| DB\_PRIVS |

| DELEGATION\_TOKENS |

| FUNCS |

| FUNC\_RU |

| GLOBAL\_PRIVS |

| HIVE\_LOCKS |

| IDXS |

| INDEX\_PARAMS |

| KEY\_CONSTRAINTS |

| MASTER\_KEYS |

| NEXT\_COMPACTION\_QUEUE\_ID |

| NEXT\_LOCK\_ID |

| NEXT\_TXN\_ID |

| NOTIFICATION\_LOG |

| NOTIFICATION\_SEQUENCE |

| NUCLEUS\_TABLES |

| PARTITIONS |

| PARTITION\_EVENTS |

| PARTITION\_KEYS |

| PARTITION\_KEY\_VALS |

| PARTITION\_PARAMS |

| PART\_COL\_PRIVS |

| PART\_COL\_STATS |

| PART\_PRIVS |

| ROLES |

| ROLE\_MAP |

| SDS |

| SD\_PARAMS |

| SEQUENCE\_TABLE |

| SERDES |

| SERDE\_PARAMS |

| SKEWED\_COL\_NAMES |

| SKEWED\_COL\_VALUE\_LOC\_MAP |

| SKEWED\_STRING\_LIST |

| SKEWED\_STRING\_LIST\_VALUES |

| SKEWED\_VALUES |

| SORT\_COLS |

| TABLE\_PARAMS |

| TAB\_COL\_STATS |

| TBLS |

| TBL\_COL\_PRIVS |

| TBL\_PRIVS |

| TXNS |

| TXN\_COMPONENTS |

| TYPES |

| TYPE\_FIELDS |

| VERSION |

| WRITE\_SET |

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

57 rows in set (0.00 sec)

## 启动 Hive

### 简单测试

**启动Hive**

cd /usr/local/apache-hive-2.3.2-bin/bin

./hive

**创建 hive 库**

hive> **create** **database** ymq;

OK

Time taken: 0.742 seconds

**选择库**

hive> **use** ymq;

OK

Time taken: 0.036 seconds

**创建表**

hive> **create** table test (mykey **string**,myval **string**);

OK

Time taken: 0.569 seconds

**插入数据**

hive> insert into test values("1","www.ymq.io");

WARNING: Hive-**on**-MR **is** deprecated **in** Hive 2 **and** may **not** be available **in** the future versions. Consider using a different execution engine (i.e. spark, tez) **or** using Hive 1.X releases.

Query ID = hadoop\_20170922011126\_abadfa44-8ebe-4ffc-9615-4241707b3c03

Total jobs = 3

Launching Job 1 **out** **of** 3

Number **of** reduce tasks **is** set **to** 0 since there's no reduce operator

Starting Job = job\_1506006892375\_0001, Tracking URL = http://node1:8088/proxy/application\_1506006892375\_0001/

Kill Command = /home/hadoop/hadoop-2.7.4//bin/hadoop job -kill job\_1506006892375\_0001

Hadoop job information **for** Stage-1: number **of** mappers: 1; number **of** reducers: 0

2017-09-22 01:12:12,763 Stage-1 **map** = 0%, reduce = 0%

2017-09-22 01:12:20,751 Stage-1 **map** = 100%, reduce = 0%, Cumulative CPU 1.24 sec

MapReduce Total cumulative CPU time: 1 seconds 240 msec

Ended Job = job\_1506006892375\_0001

Stage-4 **is** selected by condition resolver.

Stage-3 **is** filtered **out** by condition resolver.

Stage-5 **is** filtered **out** by condition resolver.

Moving data **to** directory hdfs://node1:9000/user/hive/warehouse/ymq.db/test/.hive-staging\_hive\_2017-09-22\_01-11-26\_242\_8022847052615616955-1/-ext-10000

Loading data **to** table ymq.test

MapReduce Jobs Launched:

Stage-Stage-1: **Map**: 1 Cumulative CPU: 1.24 sec HDFS Read: 4056 HDFS Write: 77 SUCCESS

Total MapReduce CPU Time Spent: 1 seconds 240 msec

OK

Time taken: 56.642 seconds

查询数据

**hive**> **select** \* **from** **test**;

**OK**

1 **www**.ymq.io

**Time** **taken**: 0.253 **seconds**, **Fetched**: 1 **row**(**s**)

### 页面数据

**在界面上查看刚刚写入的hdfs数据**







