# 大数据技术之Hue

**讲师：刘峰吉**

# 一、Hue简介

## 1、来源

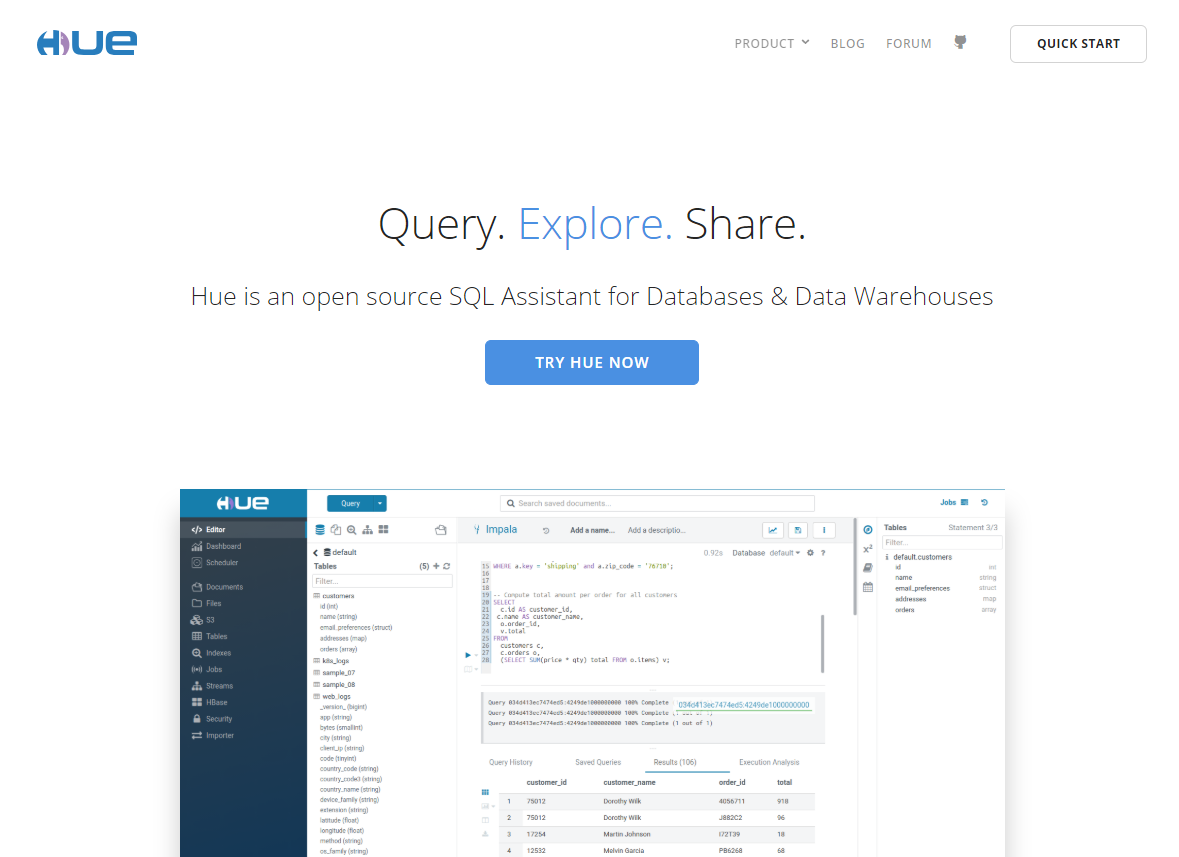
HUE=HadoopUser Experience，看这名字就知道怎么回事了吧，没错，直白来说就是Hadoop用户体验，是一个开源的Apache Hadoop UI系统，由Cloudera Desktop演化而来，最后Cloudera公司将其贡献给Apache基金会的Hadoop社区，它是基于Python Web框架Django实现的。通过使用HUE我们可以在浏览器端的Web控制台上与Hadoop集群进行交互来分析处理数据。

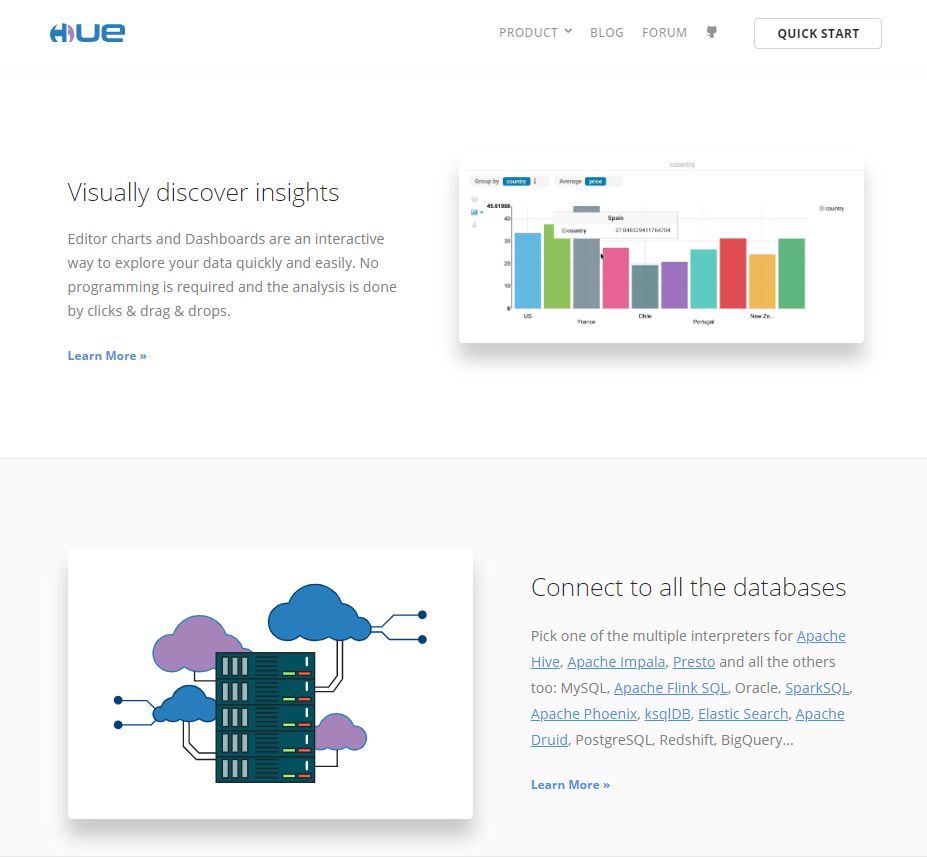
CDH 二进制安装包下载地址

http://archive.cloudera.com/cdh5/cdh/5/

## 2、官网及使用者

官网网站：http://gethue.com/

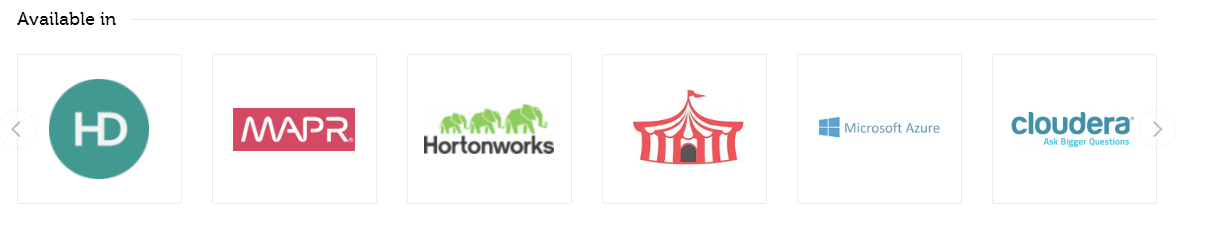




快速体验

docker run -it -p 8888:8888 gethue/hue:latest

使用的公司



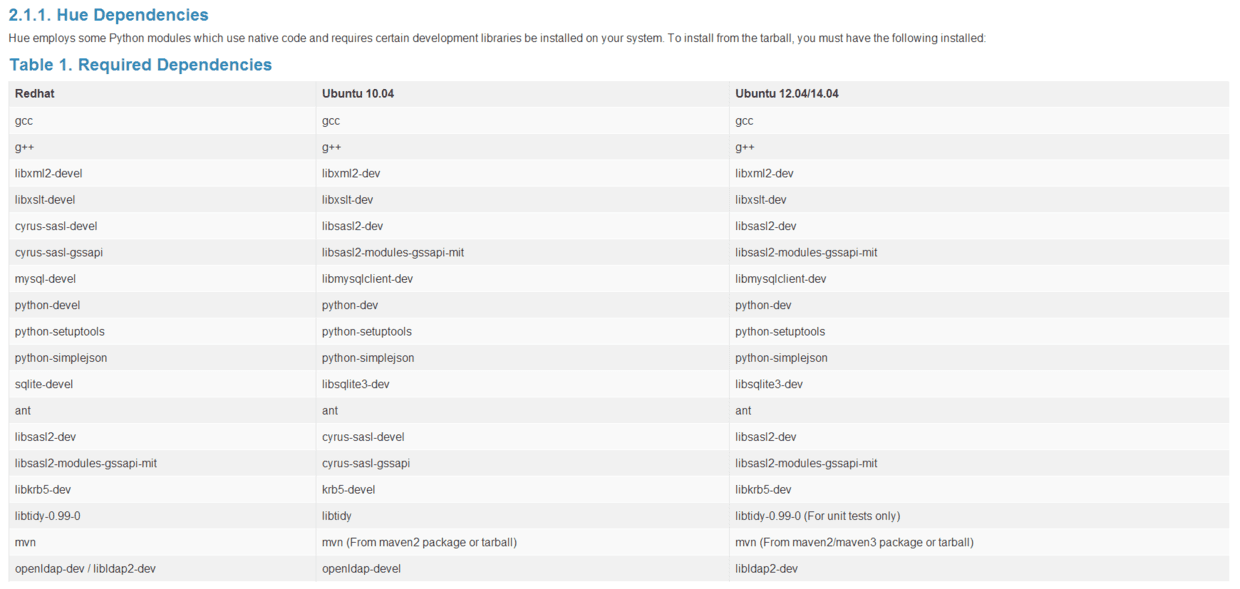
# 二、HUE安装

## 1、帮助文档

http://archive.cloudera.com/cdh5/cdh/5/hue-3.7.0-cdh5.3.0/manual.html

## 2、准备环境依赖

打开官方网站可以找到对应的部署HUE所需的各种依赖列表，如下图所示



如上图所示，这部分内容是告诉你，安装编译Hue需要依赖哪些Linux安装包，你只需要使用yum命令一次安装就可以了，在此给大家整理好该命令（注意使用root权限安装）

|  |
| --- |
| [root@hadoop102 ~]# yum -y install \  ant asciidoc cyrus-sasl-devel cyrus-sasl-gssapi cyrus-sasl-plain gcc gcc-c++ gcc\* krb5-devel libffi-devel libxml2-devel libxslt-devel make openldap-devel python-devel sqlite-devel gmp-devel |

尖叫提示：使用yum安装这些包的同时，也会自动安装openJDK的依赖，所以，请自行删除安装后的openJDK

## 3、卸载openjdk

|  |
| --- |
| [root@hadoop102 ~]# rpm -qa | grep -i java  [root@hadoop102 ~]# rpm -e --nodeps xxx-java-xxx.rpm |

## 4、升级python版本

由于HUE框架依赖python2.7，而CentOS7以下的系统使用的都是python2.6.6，并且CentOS6.8的yum也是依赖2.6，所以升级过程会稍微繁琐，特此予以讲解，Centos7可以查看Python版本，适情况而定

### 1）环境准备

#### （1）查看python版本

|  |
| --- |
| [root@hadoop102 ~]# python -V  Python 2.7.5 |

Hue 在源码编译的时候，需要python2.6.8 以上版本，而centos6.8 是2.6.6 centos7 是2.7.5，所以centos6.8 需要升级一下python

#### （2）安装GCC与wget

用于编译源码包与资源下载

|  |
| --- |
| [root@hadoop102 ~]# yum install gcc gcc-c++  [root@hadoop102 ~]# yum install wget |

#### （3）安装xz工具

用于解压tar.xz格式文件

|  |
| --- |
| [root@hadoop102 ~]# wget \  http://down1.chinaunix.net/distfiles/xz-5.0.3.tar.bz2  [root@hadoop102 ~]# tar -xjvf xz-5.0.3.tar.bz2  [root@hadoop102 ~]# cd xz-5.0.3  [root@hadoop102 ~]# ./configure  [root@hadoop102 ~]# make  [root@hadoop102 ~]# make install |

### 2）安装Python2.7

#### （1）下载Python

|  |
| --- |
| [root@hadoop102 ~]# wget \  https://www.python.org/ftp/python/2.7.11/Python-2.7.11.tar.xz |

#### （2）解压Python

|  |
| --- |
| [root@hadoop102 ~]# xz -d Python-2.7.11.tar.xz  [root@hadoop102 ~]# tar -xvf Python-2.7.11.tar |

#### （3）编译安装python

|  |
| --- |
| [root@hadoop102 ~]# cd Python-2.7.11  [root@hadoop102 ~]# ./configure  [root@hadoop102 ~]# make  [root@hadoop102 ~]# make install |

#### （4）修改系统python版本

将系统指向的python从2.6修改到2.7版本

|  |
| --- |
| [root@hadoop102 ~]# /usr/local/bin/python2.7 -V  [root@hadoop102 ~]# mv /usr/bin/python /usr/bin/python.bak  [root@hadoop102 ~]# ln -s /usr/local/bin/python2.7 /usr/bin/python |

#### （5）修改yum python版本

将yum对python的引用重新指向python2.6(即：yum使用2.6，系统用2.7)

|  |
| --- |
| [root@hadoop102 ~]# vim /usr/bin/yum  !/usr/bin/python  改为：  !/usr/bin/python2.6 |

#### （6）检查python版本

|  |
| --- |
| [root@hadoop102 ~]# python -V |

#### （7）检查yum是否可用

|  |
| --- |
| [root@hadoop102 ~]# yum install -y tree |

## 5、Centos7安装MySql

下载mysql 的yum源

因为Centos7 以后，本身不带mysql的yum源，所以需要手动下载一下

### 1）获得mysql.rpm包

获得mysql-community-release-el7-5.noarch.rpm 文件

|  |
| --- |
| [root@hadoop102 yum.repos.d]# wget \  http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm |

### 2）安装mysql.rpm包

安装mysql-community-release-el7-5.noarch.rpm 文件

|  |
| --- |
| [root@hadoop102 yum.repos.d]# rpm -ivh \  mysql-community-release-el7-5.noarch.rpm |

**尖叫提示：安装rpm文件，获得两个repo文件**

**mysql-community.repo**

**mysql-community-source.repo**

### 3）安装mysql

yum安装mysql服务(master+slave)

|  |
| --- |
| [root@hadoop102 yum.repos.d]# yum install -y mysql-server |

### 4）查看mysql都安装了什么服务

|  |
| --- |
| [root@hadoop102 module]$ rpm -qa | grep -i mysql  mysql-community-client-5.6.49-2.el7.x86\_64  mysql-community-common-5.6.49-2.el7.x86\_64  mysql-community-release-el7-5.noarch  mysql-community-libs-5.6.49-2.el7.x86\_64  mysql-community-server-5.6.49-2.el7.x86\_64 |

### 5）启动mysql服务

|  |
| --- |
| [root@hadoop102 module]# systemctl enable mysqld.service  [root@hadoop102 module]# systemctl start mysqld.service  [root@hadoop102 module]# systemctl status mysqld.service |

### 6）登陆mysql

|  |
| --- |
| [root@hadoop102 yum.repos.d]# mysql -uroot -p  Enter password: (没有密码)  mysql> set password = password('123456');  mysql> flush privileges;  mysql> exit; |

### 7）远程登陆

|  |
| --- |
| mysql> show databases;  mysql> use mysql;  mysql> show tables;  mysql> desc user;  mysql> select host,user,password from user;  mysql> update user set host = '%' where user = 'root' and host = 'localhost'; |

grant all privileges on \*.\* to 'root'@'%' identified by '123456';

## 6、解压Hue

http://archive.cloudera.com/cdh5/cdh/5/

|  |
| --- |
| [alex@hadoop102 software]$ tar -xzvf hue-3.7.0-cdh5.3.6.tar.gz \  -C ../module/cdh |

**尖叫提示:一定不能使用root用户**

## 7、编译HUE

到hue安装目录下，执行make apps

|  |
| --- |
| [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ make apps |

大概等个几分钟之后，就编译成功了。

尖叫提示：使用普通用户编译，一定不能使用root用户

## 8、编译Hue 异常解决

### 1）问题1

缺少gcc

解决办法

|  |
| --- |
| [alex@hadoop102 yum.repos.d]$ sudo yum install -y gcc\* |

### 2）问题2

OpenSSL/crypto/x509.h:17:25: fatal error: openssl/ssl.h: No such file or directory

#include <openssl/ssl.h>

解决办法

|  |
| --- |
| [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ sudo yum install -y openssl-devel |

### 3）问题3

OpenSSL/crypto/crl.c:6:23: error: static declaration of ‘X509\_REVOKED\_dup’ follows non-static declaration

static X509\_REVOKED \* X509\_REVOKED\_dup(X509\_REVOKED \*orig) {

^

In file included from /usr/include/openssl/ssl.h:156:0,

from OpenSSL/crypto/x509.h:17,

from OpenSSL/crypto/crypto.h:30,

from OpenSSL/crypto/crl.c:3:

/usr/include/openssl/x509.h:751:15: note: previous declaration of ‘X509\_REVOKED\_dup’ was here

X509\_REVOKED \*X509\_REVOKED\_dup(X509\_REVOKED \*rev);

解决办法

|  |
| --- |
| 在该文件中删除751、752 这两行  /usr/include/openssl/x509.h  [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ sudo vim /usr/include/openssl/x509.h  751 X509\_REVOKED \*X509\_REVOKED\_dup(X509\_REVOKED \*rev);  752 X509\_REQ \*X509\_REQ\_dup(X509\_REQ \*req);  ##必须删掉，注释不行 |

### 4）问题4

File "/opt/module/cdh/hue-3.7.0-cdh5.3.6/desktop/core/ext-py/MySQL-python-1.2.3c1/setup\_posix.py", line 43, in get\_config

libs = mysql\_config("libs\_r")

File "/opt/module/cdh/hue-3.7.0-cdh5.3.6/desktop/core/ext-py/MySQL-python-1.2.3c1/setup\_posix.py", line 24, in mysql\_config

raise EnvironmentError("%s not found" % (mysql\_config.path,))

EnvironmentError: mysql\_config not found

make[2]: \*\*\* [/opt/module/cdh/hue-3.7.0-cdh5.3.6/desktop/core/build/MySQL-python-1.2.3c1/egg.stamp] Error 1

make[2]: Leaving directory `/opt/module/cdh/hue-3.7.0-cdh5.3.6/desktop/core'

make[1]: \*\*\* [.recursive-env-install/core] Error 2

make[1]: Leaving directory `/opt/module/cdh/hue-3.7.0-cdh5.3.6/desktop'

make: \*\*\* [desktop] Error 2

|  |
| --- |
| [alex@hadoop102 yum.repos.d]$ sudo yum install -y mysql-devel |

### 5）问题5

Error: Package: 1:mariadb-devel-5.5.65-1.el7.x86\_64 (base)

解决办法

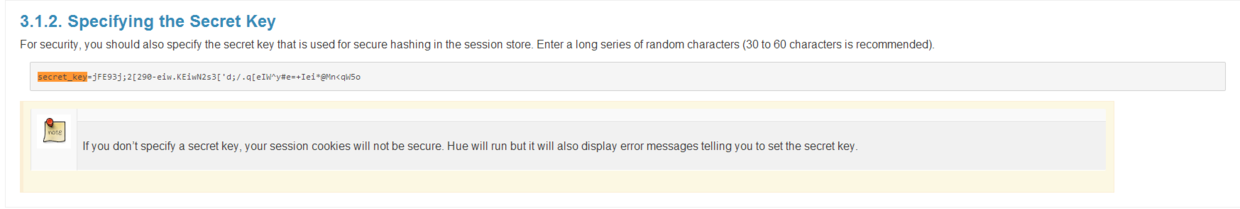
|  |
| --- |
| [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ rpm -qa | grep -i mysql  [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ sudo rpm -e --nodeps \  mysql-community-common-5.7.21-1.el7.x86\_64 \  mysql-community-client-5.7.21-1.el7.x86\_64 \  mysql-community-server-5.7.21-1.el7.x86\_64 \  mysql-community-libs-compat-5.7.21-1.el7.x86\_64 \  mysql-community-libs-5.7.21-1.el7.x86\_64  [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ rpm -qa | grep -i mysql  由于centos7 默认安装的是mariadb，没有mysql的yum源  [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ sudo cd /etc/yum.repos.d/  [alex@hadoop102 yum.repos.d]$ sudo wget \  http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm  [alex@hadoop102 yum.repos.d]$ sudo rpm -ivh mysql-community-release-el7-5.noarch.rpm  [alex@hadoop102 yum.repos.d]$ sudo yum install -y mysql-server mysql-devel  [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ make apps |

## 9、配置HUE

修改Hue.ini文件

文件位置：/opt/module/cdh/hue-3.7.0-cdh5.3.6/desktop/conf/hue.ini

其中的secret\_key请参照官方网站配置：



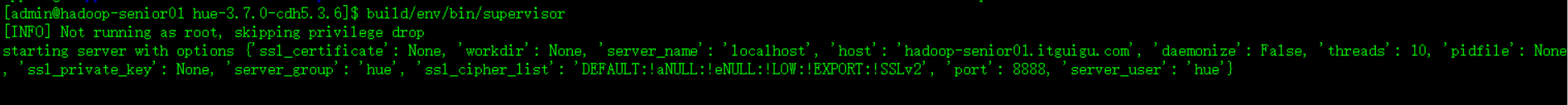
|  |
| --- |
| [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ vim desktop/conf/hue.ini  secret\_key=jFE93j;2[290-eiw.KEiwN2s3['d;/.q[eIW^y#e=+Iei\*@Mn<qW5o  http\_host=hadoop102  http\_port=8888  time\_zone=Asia/Shanghai |

## 10、启动Hue

完成之后呢，保存退出，我们来使用命令启动Hue

|  |
| --- |
| [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ chmod 755 desktop/desktop.db  [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ build/env/bin/supervisor |

出现如下界面表示启动成功

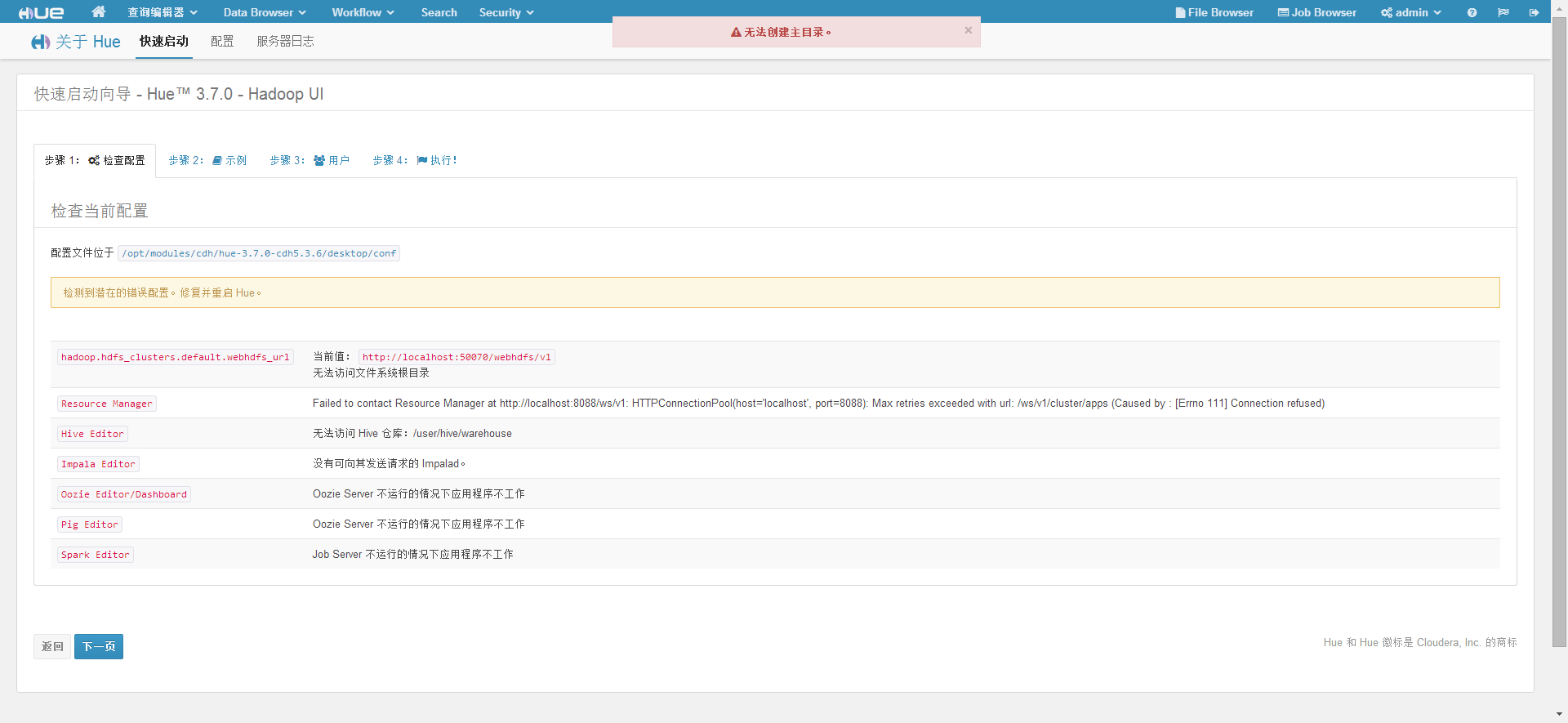


## 11、浏览器访问

|  |
| --- |
| http://hadoop102:8888 |



这句话是在提示你，第一次使用本工具，需要创建一个用户及密码，且会成为hue的超级用户凭证，在此呢，我设置为alex用户名，密码随意，那就000000吧，然后呢就可以见到如下界面了



# 三、Hue与HDFS集成

## 1、梳理集群环境

|  |  |  |
| --- | --- | --- |
| hadoop102 | hadoop103 | hadoop104 |
| NameNode |  |  |
| DataNode | DataNode | DataNode |
|  |  | SecendNameNode |
|  | ResourceManager |  |
| NodeManager | NodeManager | NodeManager |
| JobHistoryServer |  |  |
| Zookeeper | Zookeeper | Zookeeper |
| Hmaster |  |  |
| HRegionServer | HRegionServer | HRegionServer |
| Hive |  |  |
|  | MySQL |  |
| Oozie |  |  |
| Sqoop |  |  |
| Spark | Spark | Spark |

## 2、配置HDFS

### 1）hdfs-site.xml

|  |
| --- |
| <!-- 指定HDFS副本的数量 -->  <property>  <name>dfs.replication</name>  <value>3</value>  </property>  <!-- 指定secondarynamenode位置 -->  <property>  <name>dfs.namenode.secondary.http-address</name>  <value>hadoop104:50090</value>  </property>  <!-- 浏览器可以查看 http:hadoop102:50070-->  <property>  <name>dfs.webhdfs.enabled</name>  <value>true</value>  <description>Enable WebHDFS (REST API) in Namenodes and Datanodes.</description>  </property>  <!-- 关闭HDFS 检查 -->  <property>  <name>dfs.permissions.enabled</name>  <value>false</value>  </property> |

### 2）core-site.xml

|  |
| --- |
| <!-- 指定HDFS中NameNode的地址 -->  <property>  <name>fs.defaultFS</name>  <value>hdfs://hadoop102:9000</value>  </property>  <!-- 指定hadoop运行时产生文件的存储目录 -->  <property>  <name>hadoop.tmp.dir</name>  <value>/opt/module/cdh/hadoop-2.5.0-cdh5.3.6/data/tmp</value>  </property>  尖叫提示：标红处，此处是用户名  <!-- Oozie Server的Hostname -->  <!--允许哪些框架被oozie 代理用户去操作hadoop修改成自己的用户名 -->  <property>  <name>hadoop.proxyuser.alex.hosts</name>  <value>\*</value>  </property>  <!-- 允许被Oozie代理的用户组 -->  <!-- ## 允许oozie 代理用户去操作hadoop，修改成自己的用户名-->  <property>  <name>hadoop.proxyuser.alex.groups</name>  <value>\*</value>  </property>  <!-- 如果你的Hadoop配置了高可用，则必须通过httpfs来访问，需要添加如下属性，反则则不必须。（如果HUE服务与Hadoop服务不在同一节点，则必须配置）-->  <property>  <name>hadoop.proxyuser.httpfs.hosts</name>  <value>\*</value>  </property>  <property>  <name>hadoop.proxyuser.httpfs.groups</name>  <value>\*</value>  </property> |

### 3）httpfs-site.xml

|  |
| --- |
| 尖叫提示：标红处，此处就是hue,不换  解释：这两个属性主要用于HUE服务与Hadoop服务不在同一台节点上所必须的配置  <property>  <name>httpfs.proxyuser.hue.hosts</name>  <value>\*</value>  </property>  <property>  <name>httpfs.proxyuser.hue.groups</name>  <value>\*</value>  </property> |

尖叫提示：

如果没有配置NameNode的HA，HUE可以用WebHDFS来管理HDFS

如果配置了NameNodeHA，则HUE只可用HttpFS来管理HDFS

### 4）yarn-site.xml

|  |
| --- |
| <!-- reducer获取数据的方式 -->  <property>  <name>yarn.nodemanager.aux-services</name>  <value>mapreduce\_shuffle</value>  </property>  <!-- 指定YARN的ResourceManager的地址 -->  <property>  <name>yarn.resourcemanager.hostname</name>  <value>hadoop103</value>  </property>  <property>  <name>yarn.log-aggregation-enable</name>  <value>true</value>  </property>  <property>  <name>yarn.log-aggregation.retain-seconds</name>  <value>604800</value>  </property>  <!-- 任务历史服务 -->  <property>  <name>yarn.log.server.url</name>  <value>http://hadoop102:19888/jobhistory/logs/</value>  </property> |

### 5）map-site.xml

|  |
| --- |
| <!-- 指定mr运行在yarn上 -->  <property>  <name>mapreduce.framework.name</name>  <value>yarn</value>  </property>  <!-- JVM重用 -->  <property>  <name>mapreduce.job.jvm.numtasks</name>  <value>10</value>  <description>How many tasks to run per jvm. If set to -1, there is no limit. </description>  </property>  <!-- 配置 MapReduce JobHistory Server 地址 ，默认端口10020 -->  <property>  <name>mapreduce.jobhistory.address</name>  <value>hadoop102:10020</value>  </property>  <!-- 配置 MapReduce JobHistory Server web ui 地址， 默认端口19888 -->  <property>  <name>mapreduce.jobhistory.webapp.address</name>  <value>hadoop102:19888</value>  </property> |

### 6）other

|  |
| --- |
| [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ vim etc/hadoop/hadoop-env.sh  export JAVA\_HOME=/opt/module/jdk1.8.0\_144  [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ vim etc/hadoop/mapred-env.sh  export JAVA\_HOME=/opt/module/jdk1.8.0\_144  [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ vim etc/hadoop/yarn-env.sh  export JAVA\_HOME=/opt/module/jdk1.8.0\_144  [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ vim etc/hadoop/slaves  hadoop102  hadoop103  hadoop104 |

## 3、分发同步配置

|  |
| --- |
| [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ xsync etc/ |

## 4、启动hadoop httpfs服务

|  |
| --- |
| [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ sbin/httpfs.sh start |

## 5、启动HDFS服务

|  |
| --- |
| [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ sbin/start-dfs.sh |

## 6、启动Yarn服务

|  |
| --- |
| [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ sbin/start-yarn.sh |

## 7、启动HistoryServer服务

|  |
| --- |
| [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ sbin/mr-jobhistory-daemon.sh \  start historyserver |

## 8、配置hue.ini文件

找到[hdfs\_clusters]标签

|  |
| --- |
| [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ vim desktop/conf/hue.ini  [[hdfs\_clusters]]  # HA support by using HttpFs  [[[default]]]  # Enter the filesystem uri  fs\_defaultfs=hdfs://hadoop102:9000  # fs\_defaultfs=hdfs://mycluster  # NameNode logical name.  # 如果开启了高可用，需要配置如下  ## logical\_name=mycluster  # Use WebHdfs/HttpFs as the communication mechanism.  # Domain should be the NameNode or HttpFs host.  # Default port is 14000 for HttpFs.  ## webhdfs\_url=http://localhost:50070/webhdfs/v1  webhdfs\_url=http://hadoop102:14000/webhdfs/v1  # Change this if your HDFS cluster is Kerberos-secured  ## security\_enabled=false  # Default umask for file and directory creation, specified in an octal value.  ## umask=022  # Directory of the Hadoop configuration  ## hadoop\_conf\_dir=$HADOOP\_CONF\_DIR when set or '/etc/hadoop/conf'  hadoop\_conf\_dir=/opt/module/cdh/hadoop-2.5.0-cdh5.3.6/etc/hadoop  hadoop\_hdfs\_home=/opt/module/cdh/hadoop-2.5.0-cdh5.3.6  hadoop\_bin=/opt/module/cdh/hadoop-2.5.0-cdh5.3.6/bin |

## 9、测试

开启HUE服务

|  |
| --- |
| [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ build/env/bin/supervisor |

打开HUE的页面，进行HDFS管理 http://hadoop102:8888

尖叫提示：如果提示错误根目录应该归属于hdfs，请修改python变量，位置如下

|  |
| --- |
| /opt/module/cdh/hue-3.7.0-cdh5.3.6/desktop/libs/hadoop/src/hadoop/fs/webhdfs.py  [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ vim desktop/libs/hadoop/src/hadoop/fs/webhdfs.py  修改其中的变量值为  DEFAULT\_HDFS\_SUPERUSER = '**alex**'  标红处：此处为用户名 alex |

然后重启HUE服务即可

尖叫提示：

启动HUE服务时，请先kill掉之前的HUE服务，如果提示地址被占用，请使用如下命令查看占用8888端口的进程并kill掉

|  |
| --- |
| [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ netstat -tunlp | grep 8888 |

# 四、Hue与Yarn集成

## 1、配置hue.ini文件

找到[[yarn\_clusters]]标签，涉及修改配置如下

|  |
| --- |
| [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ vim desktop/conf/hue.ini  [[yarn\_clusters]]  [[[default]]]  resourcemanager\_host=hadoop103  resourcemanager\_port=8032  submit\_to=True  logical\_name=cluster-yarn1(如果开高可用的话，配置这一行)  resourcemanager\_api\_url=http://hadoop103:8088  proxy\_api\_url=http://hadoop103:8088  history\_server\_api\_url=http://hadoop102:19888 |

## 2、重启Hue测试

|  |
| --- |
| [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ build/env/bin/supervisor |

## 3、测试

|  |
| --- |
| [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ bin/yarn jar \  share/hadoop/mapreduce/hadoop-mapreduce-examples-2.5.0.jar \  wordcount /wcinput/ /wcoutput/ |

尖叫提示：因为启动hdfs服务，这里使用的是hdfs 上面的路径

# 五、Hue与Hive集成

## 1、解压hive

|  |
| --- |
| [alex@hadoop102 software]$ tar -xzvf cdh/hive-0.13.1-cdh5.3.6.tar.gz \  -C ../module/cdh/ |

## 2、修改hive-env.sh

|  |
| --- |
| [alex@hadoop102 hive-0.13.1-cdh5.3.6]$ vim conf/hive-env.sh  export JAVA\_HOME=/opt/module/jdk1.8.0\_144  export HADOOP\_HOME=/opt/module/cdh/hadoop-2.5.0-cdh5.3.6  export HIVE\_CONF\_DIR=/opt/module/cdh/hive-0.13.1-cdh5.3.6/conf |

## 3、hive-log4j.properties

|  |
| --- |
| [alex@hadoop102 hive-0.13.1-cdh5.3.6]$ vim conf/hive-log4j.properties  hive.log.dir=/opt/module/cdh/hive-0.13.1-cdh5.3.6/logs |

## 4、hdfs创建目录

|  |
| --- |
| [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ bin/hadoop fs \  -mkdir /tmp  [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ bin/hadoop fs \  -mkdir -p /user/hive/warehouse  [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ bin/hadoop fs \  -chmod g+w /tmp  [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ bin/hadoop fs \  -chmod g+w /user/hive/warehouse |

## 5、添加mysql驱动包

|  |
| --- |
| [alex@hadoop102 cdh]$ sudo chown alex:alex \  mysql-connector-java-5.1.27-bin.jar  [alex@hadoop102 cdh]$ chmod 755 mysql-connector-java-5.1.27-bin.jar  [alex@hadoop102 cdh]$ cp mysql-connector-java-5.1.27-bin.jar \  /opt/module/cdh/hive-0.13.1-cdh5.3.6/lib/ |

## 6、hive-site.xml

修改Hive配置文件hive-site.xml

HUE与hive集成需要hive开启HiveServer2服务，相关配置如下

|  |
| --- |
| [alex@hadoop102 hive-0.13.1-cdh5.3.6]$ vim conf/hive-site.xml  <?xml version="1.0"?>  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  <configuration>  <property>  <name>javax.jdo.option.ConnectionURL</name>  <value>jdbc:mysql://hadoop103:3306/metastore?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>root</value>  <description>username to use against metastore database</description>  </property>  <property>  <name>javax.jdo.option.ConnectionPassword</name>  <value>000000</value>  <description>password to use against metastore database</description>  </property>  <property>  <name>hive.server2.thrift.port</name>  <value>10000</value>  </property>  <property>  <name>hive.server2.thrift.bind.host</name>  <value>hadoop102</value>  </property>  <!-- 尖叫提示：把L去掉，不然报错-->  <property>  <name>hive.server2.long.polling.timeout</name>  <value>5000</value>  </property>  <property>  <name>hive.metastore.uris</name>  <value>thrift://hadoop102:9083</value>  </property>  </configuration> |

## 7、启动Hive相关服务

|  |
| --- |
| [alex@hadoop102 ~]$ sudo systemctl start mysqld.service  [alex@hadoop102 hive-0.13.1-cdh5.3.6]$ bin/hive --service metastore &  [alex@hadoop102 hive-0.13.1-cdh5.3.6]$ bin/hive --service hiveserver2 &  [alex@hadoop102 hive-0.13.1-cdh5.3.6]$ bin/beeline  beeline> !connect jdbc:hive2://hadoop102:10000  0:jdbc:hive2://hadoop102:10000> show tables;  0:jdbc:hive2://hadoop102:10000> select \* from t1; |

尖叫提示：如果设置了uris，在今后使用Hive时，那么必须启动如上两个命令，否则Hive无法正常启动。

## 8、配置hue.ini

找到[beeswax]属性标签，涉及修改如下

|  |
| --- |
| [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ vim desktop/conf/hue.ini  [beeswax]  hive\_server\_host=hadoop102  hive\_server\_port=10000  hive\_conf\_dir=/opt/module/cdh/hive-0.13.1-cdh5.3.6/conf |

## 9、重启HUE测试

|  |
| --- |
| [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ build/env/bin/supervisor |

# 六、Hue与MySql集成

## 1、配置hue.ini

找到[[[mysql]]]标签，涉及修改如下

|  |
| --- |
| [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ vim desktop/conf/hue.ini  ## [[[mysql]]]  nice\_name="ZMysqlDB"  engine=mysql  host=hadoop102  port=3306  user=root  password=123456 |

尖叫提示：[[[mysql]]] 前面的## 一定要去掉

## 2、重启HUE测试

|  |
| --- |
| [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ build/env/bin/supervisor |

如果发生异常，初始化一下数据库

|  |
| --- |
| [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ bin/hue syncdb  [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ bin/hue migrate |

# 七、Hue与Oozie集成

## 1、Oozie安装部署

### 1）解压Oozie

|  |
| --- |
| [alex@hadoop102 software]$ tar -xzvf oozie-4.0.0-cdh5.3.6.tar.gz -C /opt/module/cdh |

### 2）修改Hadoop配置

oozie是有用户的，默认用户就是oozie

#### （1）core-site.xml

|  |
| --- |
| <!-- Oozie Server的Hostname -->  ## 允许哪些框架被oozie 代理用户去操作hadoop修改成自己的用户名  <property>  <name>hadoop.proxyuser.alex.hosts</name>  <value>\*</value>  </property>  <!-- 允许被Oozie代理的用户组 -->  ## 允许oozie 代理用户去操作hadoop，修改成自己的用户名  <property>  <name>hadoop.proxyuser.alex.groups</name>  <value>\*</value>  </property> |

#### （2）mapred-site.xml

使用oozie框架，一定要配置jobhistory server

|  |
| --- |
| <!-- 配置 MapReduce JobHistory Server 地址 ，默认端口10020 -->  <property>  <name>mapreduce.jobhistory.address</name>  <value>hadoop102:10020</value>  </property>  <!-- 配置 MapReduce JobHistory Server web ui 地址， 默认端口19888 -->  <property>  <name>mapreduce.jobhistory.webapp.address</name>  <value>hadoop102:19888</value>  </property> |

#### （3）yarn-site.xml

|  |
| --- |
| <!-- 任务历史服务 -->  <property>  <name>yarn.log.server.url</name>  <value>http://hadoop102:19888/jobhistory/logs/</value>  </property> |

完成后：记得scp同步到其他机器节点

### 3）重启Hadoop集群

|  |
| --- |
| [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ sbin/start-dfs.sh  [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ sbin/start-yarn.sh  [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ sbin/mr-jobhistory-daemon.sh \  start historyserver |

尖叫提示：需要开启JobHistoryServer, 最好执行一个MR任务进行测试

### 4）解压hadooplibs

在oozie根目录下解压hadooplibs

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ tar -xzvf \  oozie-hadooplibs-4.0.0-cdh5.3.6.tar.gz -C ../ |

尖叫提示:完成后Oozie目录下会出现hadooplibs目录,目的是把解压后的hadooplibs文件夹，放到oozie目录下

### 5）创建libext

在Oozie目录下创建libext目录

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ mkdir libext |

### 6）拷贝一些依赖的Jar包

#### （1）拷贝hadooplib jar包

将hadooplibs里面的jar包，拷贝到libext目录下

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ cp -ra \  hadooplibs/hadooplib-2.5.0-cdh5.3.6.oozie-4.0.0-cdh5.3.6/\* libext/ |

#### （2）拷贝mysql驱动包

拷贝Mysql驱动包到libext目录下

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ cp -ra \  mysql-connector-java-5.1.27-bin.jar libext/ |

oozie 的元数据保存在mysql中，默认derby

### 7）拷贝ext-2.2.zip

将ext-2.2.zip拷贝到libext/目录下，ext是一个js框架，用于展示oozie前端页面

oozie是一个定时的任务调度框架，是有web页面的，依赖ext-2.2.zip

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ cp -r ext-2.2.zip libext/ |

### 8）修改Oozie配置文件

oozie-site.xml

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ vim conf/oozie-site.xml  <!-- JDBC的驱动-->  <property>  <name>oozie.service.JPAService.jdbc.driver</name>  <value>com.mysql.jdbc.Driver</value>  </property>  <!-- oozie所需的数据库地址-->  <property>  <name>oozie.service.JPAService.jdbc.url</name>  <value>jdbc:mysql://hadoop102:3306/oozie</value>  </property>  <!-- oozie所需的数据库用户名-->  <property>  <name>oozie.service.JPAService.jdbc.username</name>  <value>root</value>  </property>  <!-- oozie所需的数据库密码-->  <property>  <name>oozie.service.JPAService.jdbc.password</name>  <value>123456</value>  </property>  <!--让Oozie引用Hadoop的配置文件\*=不能删-->  <property>  <name>oozie.service.HadoopAccessorService.hadoop.configurations</name>  <value>\*=/opt/module/cdh/hadoop-2.5.0-cdh5.3.6/etc/hadoop</value>  </property> |

### 9）在Mysql中创建Oozie的数据库

进入Mysql并创建oozie数据库

|  |
| --- |
| [alex@hadoop102 ~]$ mysql -uroot -p123456  mysql > create database oozie; |

### 10）初始化Oozie

#### （1）上传yarn.tar.gz

上传Oozie目录下的yarn.tar.gz文件到HDFS

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ bin/oozie-setup.sh sharelib \  create -fs \  hdfs://hadoop102:9000 -locallib oozie-sharelib-4.0.0-cdh5.3.6-yarn.tar.gz |

尖叫提示：yarn.tar.gz文件会自行解压

命令的意思是把这个oozie-sharelib-4.0.0-cdh5.3.6-yarn.tar.gz 包解压出来，放到hdfs上面

以wordcount案例，oozie作为定时调度框架，必须先把wc.jar上传到hdfs，oozie才有能力去调度它，放到本地是不行的，执行成功之后，去50070检查对应目录有没有文件生成。

#### （2）创建oozie.sql文件

操作的时候mysql一定要有oozie这个数据库

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ bin/oozie-setup.sh db \  create -run -sqlfile oozie.sql |

尖叫提示：这个操作是在oozie这个库里面，创建了一系列的表

#### （3）打包项目，生成war包

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ bin/oozie-setup.sh prepare-war |

### 11）启动Oozie服务（附关闭Oozie服务）

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ bin/oozied.sh start  如需正常关闭Oozie服务，请使用  [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ bin/oozied.sh stop |

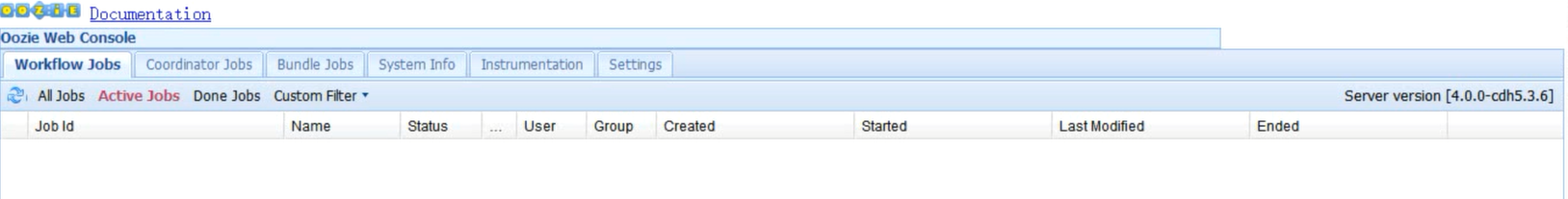
尖叫提示：[alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ bin/oozie-start.sh

WARN: Use of this script is deprecated; use 'oozied.sh start' instead

### 12）访问Oozie的Web页面

看到的web页面， 依赖js框架

|  |
| --- |
| http://hadoop102:11000/oozie |



## 1）配置hue.ini

找到[liboozie]标签以及[oozie]标签涉及修改如下

|  |
| --- |
| [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ vim desktop/conf/hue.ini  [liboozie]  oozie\_url=http://hadoop102:11000/oozie  remote\_deployement\_dir=/user/alex/oozie-apps  [oozie]  local\_data\_dir=/opt/module/cdh/oozie-4.0.0-cdh5.3.6/examples  sample\_data\_dir=/opt/module/cdh/oozie-4.0.0-cdh5.3.6/oozie-apps  # hdfs 路径  remote\_data\_dir=/user/alex/oozie-apps  # 是否允许周期性调度  enable\_cron\_scheduling=true |

## 2、启动Oozie相关服务（提示）

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ bin/oozied.sh start |

## 3、重启HUE测试

|  |
| --- |
| [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ build/env/bin/supervisor |

尖叫提示：如果提示无法关联oozie的share/lib，请使用hdfs命令创建该目录即可

|  |
| --- |
| [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ bin/hdfs dfs \  -mkdir -p /user/oozie/share/lib |

# 八、Hue与Zookeeper集成

## 1、Zookeeper安装部署

### 1）解压

|  |
| --- |
| [alex@hadoop102 software]$ tar -xzvf cdh/zookeeper-3.4.5-cdh5.3.6.tar.gz \  -C ../module/cdh/ |

### 2）创建zkdata

|  |
| --- |
| [alex@hadoop102 zookeeper-3.4.5-cdh5.3.6]$ mkdir zkdata |

### 3）修改zoo.cfg配置文件

|  |
| --- |
| [alex@hadoop102 zookeeper-3.4.5-cdh5.3.6]$ vim conf/zoo.cfg  dataDir=/opt/module/cdh/zookeeper-3.4.5-cdh5.3.6/zkdata  clientPort=2181  server.1=hadoop102:2888:3888  server.2=hadoop103:2888:3888  server.3=hadoop104:2888:3888 |

### 4）编写myid

|  |
| --- |
| [alex@hadoop102 zookeeper-3.4.5-cdh5.3.6]$ touch zkdata/myid  [alex@hadoop102 zookeeper-3.4.5-cdh5.3.6]$ echo 1 > zkdata/myid  [alex@hadoop102 zookeeper-3.4.5-cdh5.3.6]$ cat 1 > zkdata/myid  1 |

### 5）分发

|  |
| --- |
| [alex@hadoop102 cdh]$ xsync zookeeper-3.4.5-cdh5.3.6/ |

### 6）修改myid

|  |
| --- |
| [alex@hadoop103 zookeeper-3.4.5-cdh5.3.6]$ echo 1 > zkdata/myid  [alex@hadoop103 zookeeper-3.4.5-cdh5.3.6]$ cat 1 > zkdata/myid  1  [alex@hadoop104 zookeeper-3.4.5-cdh5.3.6]$ echo 1 > zkdata/myid  [alex@hadoop104 zookeeper-3.4.5-cdh5.3.6]$ cat 1 > zkdata/myid  1 |

### 7）启动zookeeper

|  |
| --- |
| [alex@hadoop102 zookeeper-3.4.5-cdh5.3.6]$ bin/zkServer.sh start  [alex@hadoop103 zookeeper-3.4.5-cdh5.3.6]$ bin/zkServer.sh start  [alex@hadoop104 zookeeper-3.4.5-cdh5.3.6]$ bin/zkServer.sh start |

## 2、配置hue.ini

找到[zookeeper]标签，涉及修改如下

|  |
| --- |
| [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ vim desktop/conf/hue.ini  [zookeeper]  [[clusters]]  [[[default]]]  host\_ports=hadoop102:2181,hadoop103:2181,hadoop104:2181 |

## 3、启动zookeeper

|  |
| --- |
| [alex@hadoop102 zookeeper-3.4.5-cdh5.3.6]$ bin/zkServer.sh start |

## 4、重启HUE测试

|  |
| --- |
| [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ build/env/bin/supervisor |

# 九、Hue与HBase集成

## 1、安装部署hbase

### 1）解压HBase到指定目录

|  |
| --- |
| [alex@hadoop102 software]$ tar -xzvf cdh/hbase-0.98.6-cdh5.3.6.tar.gz -C ../module/cdh/ |

### 2）HBase的配置文件

#### （1）hbase-env.sh

|  |
| --- |
| [alex@hadoop102 hbase-0.98.6-cdh5.3.6]$ vim conf/hbase-env.sh  export JAVA\_HOME=/opt/module/jdk1.8.0\_144  export HBASE\_MANAGES\_ZK=false  export HBASE\_CLASSPATH=/opt/module/cdh/hbase-0.98.6-cdh5.3.6/conf  export HBASE\_LOG\_DIR=/opt/module/cdh/hbase-0.98.6-cdh5.3.6 /logs |

**尖叫提示：如果使用的是JDK8及以上版本，则应在hbase-evn.sh中移除“HBASE\_MASTER\_OPTS”和“HBASE\_REGIONSERVER\_OPTS”配置。**

#### （2）hbase-site.xml

|  |
| --- |
| [alex@hadoop102 hbase-0.98.6-cdh5.3.6]$ vim conf/hbase-site.xml  <configuration>  <!-- hbase存放在hdfs上的路径，端口要和hadoop的fs.defaultFS端口一致 -->  <property>  <name>hbase.rootdir</name>  <value>hdfs://hadoop102:9000/hbase</value>  <description> </description>  </property>  <!-- 是否分布式部署 -->  <property>  <name>hbase.cluster.distributed</name>  <value>true</value>  </property>  <!-- 0.98后的新变动，之前版本没有.port,低版本默认端口为60000 -->  <property>  <name>hbase.master.port</name>  <value>16000</value>  </property>  <!-- Zookeeper 节点列表 -->  <property>  <name>hbase.zookeeper.quorum</name>  <value>hadoop102,hadoop103,hadoop104</value>  <!-- 只配置主机名就可以-->  </property>  <property>  <name>hbase.zookeeper.property.clientPort</name>  <value>2181</value>  </property>  <!-- Zookeeper 存储位置 -->  <property>  <name>hbase.zookeeper.property.dataDir</name>  <value>/opt/module/cdh/zookeeper-3.4.5-cdh5.3.6/zkdata</value>  </property>  <property>  <name>zookeeper.znode.parent</name>  <value>/hbase</value>  <description>Root ZNode for HBase in ZooKeeper. All of HBase's ZooKeeper  files that are configured with a relative path will go under this node.  By default, all of HBase's ZooKeeper file path are configured with a  relative path, so they will all go under this directory unless changed.</description>  </property>  </configuration> |

#### （3）regionservers

|  |
| --- |
| [alex@hadoop102 hbase-0.98.6-cdh5.3.6]$ vim conf/regionservers  hadoop102  hadoop103  hadoop104 |

#### （4）log4j.properties

|  |
| --- |
| [alex@hadoop102 hbase-0.98.6-cdh5.3.6]$ vim conf/log4j.properties  #hbase.log.dir=.  hbase.log.dir=/opt/module/cdh/hbase-0.98.6-cdh5.3.6/logs |

### 3）分发

|  |
| --- |
| [alex@hadoop102 cdh]$ xsync hbase-0.98.6-cdh5.3.6/ |

## 2、修改hue.ini配置

找到[hbase]标签，涉及修改内容如下

|  |
| --- |
| [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ vim desktop/conf/hue.ini  [hbase]  hbase\_clusters=(Cluster|hadoop102:9090)  hbase\_conf\_dir=/opt/module/cdh/hbase-0.98.6-cdh5.3.6/conf |

## 3、修改hbase-site.xml

|  |
| --- |
| [alex@hadoop102 hbase-0.98.6-cdh5.3.6]$ vim conf/hbase-site.xml  <property>  <name>hbase.thrift.support.proxyuser</name>  <value>true</value>  </property>  <property>  <name>hbase.regionserver.thrift.http</name>  <value>true</value>  </property> |

## 4、启动HBase的thrift服务

|  |
| --- |
| [alex@hadoop102 hbase-0.98.6-cdh5.3.6]$ bin/hbase-daemon.sh start thrift  [alex@hadoop102 hbase-0.98.6-cdh5.3.6]$ bin/start-hbase.sh |

## 5、重启HUE进行测试

|  |
| --- |
| [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ build/env/bin/supervisor |

# 十、Hue与Sqoop2集成

尖叫提示：HUE只支持Sqoop2的集成，不支持Sqoop1，在此不再演示。

http://archive.cloudera.com/cdh5/cdh/5/hue-3.7.0-cdh5.3.6/user-guide/sqoop.html

首先，说下博主我，为什么，好端端的Sqoop1用的好好的，然后又安装和学习Sqoop2呢？

因为，在Cloudera Hue里的Sqoop，是需要Sqoop2。

## 1、Sqoop2安装简介

sqoop2的安装分为server端和client端。

server端：负责与hadoop集群通信进行数据的迁移，client端负责与用户和server交互。

client端：不用安装，只需要将其安装包上传到集群中任何的一台机器上去，然后对其进行解压即可， 无需其它额外的配置。

Sqoop分client和server，server安装在Hadoop或Spark集群中的某个节点上，这个节点充当要连接sqoop的入口节点，client端不需要安装hadoop。

本博文是个入门，即只在bigdatamaster（hadoop102）上安装server端。当然你也可以假设认为server和client都在bigdatamaster（hadoop102）上

对于Sqoop和hive这样的组件，我一般都是安装在master节点，即在本博客里是bigdatamaster（hadoop102）。

http://archive.cloudera.com/cdh5/cdh/5/sqoop2-1.99.4-cdh5.3.6.tar.gz

## 2、Sqoop2 配置环境变量

|  |
| --- |
| [root@hadoop102 ~]# vim /etc/profile  #sqoop2  export SQOOP\_HOME=/opt/module/cdh/sqoop2-1.99.4-cdh5.3.6  export PATH=$PATH:$SQOOP\_HOME/bin  export CATALINA\_BASE=/opt/module/cdh/sqoop2-1.99.4-cdh5.3.6/server  export LOGDIR=$SQOOP\_HOME/logs/  #sqoop1  #export SQOOP\_HOME=/opt/module/sqoop-1.4.7.bin\_hadoop-2.6.0  #export PATH=$PATH:$SQOOP\_HOME/bin |

## 3、对于Sqoop2的server端安装配置

### 1）解压

|  |
| --- |
| [alex@hadoop102 software]$ tar -xzvf sqoop2-1.99.4-cdh5.3.6.tar.gz \  -C ../module/cdh |

### 2）配置环境变量

|  |
| --- |
| [alex@hadoop102 ~]$ sudo vim /etc/profile  #sqoop2  export SQOOP\_HOME=/opt/module/cdh/sqoop2-1.99.4-cdh5.3.6  export PATH=$PATH:$SQOOP\_HOME/bin  export CATALINA\_BASE=/opt/module/cdh/sqoop2-1.99.4-cdh5.3.6/server  export LOGDIR=$SQOOP\_HOME/logs/  [alex@hadoop102 ~]$ source /etc/profile  [alex@hadoop102 sqoop2-1.99.4-cdh5.3.6]$ sqoop.sh version  Sqoop home directory: /opt/module/cdh/sqoop2-1.99.4-cdh5.3.6  Command is not recognized. |

### 3）修改sqoop配置sqoop.properties

|  |
| --- |
| [alex@hadoop102 sqoop2-1.99.4-cdh5.3.6]$ vim server/conf/sqoop.properties  **# Hadoop configuration directory**  **org.apache.sqoop.submission.engine.mapreduce.configuration.directory=/opt/module/cdh/hadoop-2.5.0-cdh5.3.6/etc/hadoop** |

将 org.apache.sqoop.submission.engine.mapreduce.configuration.directory后面 hadoop的位置修改为自己安装的hadoop配置文件位置

例如：/opt/module/cdh/hadoop-2.5.0-cdh5.3.6

### 4）修改sqoop配置catalina.properties

vi /sqoop/server/conf/catalina.properties

|  |
| --- |
| [alex@hadoop102 sqoop2-1.99.4-cdh5.3.6]$ vim \  server/conf/catalina.properties |

修改sqoop读取hadoop的jar包的路径

将common.loader行后的/usr/lib/hadoop/lib/.jar改成自己的hadoop jar 包目录

/opt/module/cdh/hadoop-2.5.0-cdh5.3.6/share/hadoop/common/\*.jar,

/opt/module/cdh/hadoop-2.5.0-cdh5.3.6/share/hadoop/common/lib/\*.jar,

/opt/module/cdh/hadoop-2.5.0-cdh5.3.6/share/hadoop/hdfs/\*.jar,

/opt/module/cdh/hadoop-2.5.0-cdh5.3.6/share/hadoop/hdfs/lib/\*.jar,

/opt/module/cdh/hadoop-2.5.0-cdh5.3.6/share/hadoop/mapreduce/\*.jar,

/opt/module/cdh/hadoop-2.5.0-cdh5.3.6/share/hadoop/mapreduce/lib/\*.jar,

/opt/module/cdh/hadoop-2.5.0-cdh5.3.6/share/hadoop/tools/\*.jar,

/opt/module/cdh/hadoop-2.5.0-cdh5.3.6/share/hadoop/tools/lib/\*.jar,

/opt/module/cdh/hadoop-2.5.0-cdh5.3.6/share/hadoop/yarn/\*.jar,

/opt/module/cdh/hadoop-2.5.0-cdh5.3.6/share/hadoop/yarn/lib/\*.jar

**注意： 在修改common.loader的过程中， 不能换行写到一行,而且这是一个追加的操作**

本步骤的另外的一种方法是： 直接将上诉的包 拷贝到$SQOOP\_HOME/server/lib文件夹内部- 将mysql的连接jar包拷贝的$SQOOP\_HOME/lib文件夹中（lib文件夹需要自己创建）到此sqoop就基本配置完成可以直接运行.

$SQOOP\_HOME/server/conf/server.xml是Tomcat的配置文件，端口什么的可以在这个文件设置这里，我暂时不设置。

复制mysql-connector-java-5.1.21.jar到$SQOOP\_HOME/server/lib/下

## 4、启动 sqoop2 的 Server

|  |
| --- |
| [alex@hadoop102 sqoop2-1.99.4-cdh5.3.6]$ bin/sqoop.sh server start |

## 5、启动Sqoop2的client

我这里做个最简单的，在bigdatamaster（hadoop102）节点上，进入客户端交互模式，进入sqoop控制台

|  |
| --- |
| [alex@hadoop102 sqoop2-1.99.4-cdh5.3.6]$ bin/sqoop.sh client  Sqoop home directory: /opt/module/sqoop2-1.99.4-cdh5.3.6  May 07, 2020 8:26:47 PM java.util.prefs.FileSystemPreferences$1 run  INFO: Created user preferences directory.  Sqoop Shell: Type 'help' or '\h' for help.  sqoop:000> |

## 6、Sqoop2连接服务器

在Sqoop2的client连接Sqoop2的server

|  |
| --- |
| sqoop:000> set server --host hadoop102 --port 12000 --webapp sqoop  sqoop:000> set server --host localhost --port 12000 --webapp sqoop  sqoop:000> show version --all |

当看到show version -all正确的显示 就说明了Sqoop2的client连接上了Sqoop2的服务器。

show version --all显示服务器、客户端的版本信息

如果server显示错误，重启一下 server./sqoop.sh server stop

## 7、常用命令

|  |
| --- |
| sqoop:000> show connector --all 查看连接器  sqoop:000> show connection --all 查看连接  sqoop:000> show connection --xid 1 查看id为1的连接  sqoop:000> create connection --cid 1 创建id为1的连接  <pre name="code" class="java">Creating connection for connector with id 1  Please fill following values to create new connection object  Name: mysql --输入名称    Connection configuration    JDBC Driver Class: com.mysql.jdbc.Driver --输入  JDBC Connection String: jdbc:mysql://bigdatamaster:3306/sqoop --输入  Username: root --输入  Password: \*\*\*\*\*\* --输入  JDBC Connection Properties:  There are currently 0 values in the map:  entry#    Security related configuration options    Max connections: 20 --输入  New connection was successfully created with validation status FINE and persistent id 1  sqoop:000> create job --xid 1 --type import  Creating job for connection with id 1  Please fill following values to create new job object  Name: mysql\_job    Database configuration    Schema name:  Table name: userinfo 要全量导出一张表，请填写表名，table name 和 table sql statement不能同时配置  Table SQL statement: 如果填写格式必须为 select \* from userinfo where ${CONDITIONS}  Table column names:  Partition column name: id 使用哪个字段来填充过滤条件 userid  Nulls in partition column:  Boundary query: 如果选择sql方式，这里要写一个查询语句，返回值需为整形，sqoop运行job时，会自动填充${CONDITIONS} 这个占位符,如：select 0,3 from userinfo      Output configuration    Storage type:  : HDFS  Choose: 0  Output format:  : TEXT\_FILE  : SEQUENCE\_FILE  Choose: 1  Compression format:  : NONE  : DEFAULT  : DEFLATE  : GZIP  : BZIP2  : LZO  : LZ4  : SNAPPY  Choose: 0  Output directory: /home/hadoop/out    Throttling resources    Extractors:  Loaders:  New job was successfully created with validation status FINE and persistent id 1  sqoop:000> start job --jid 1 启动  Submission details  Job ID: 1  Server URL: http://localhost:12000/sqoop/  [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ bin/hadoop fs -ls /mysql/out |

## 8、修改hue.ini配置

找到[hbase]标签，涉及修改内容如下

|  |
| --- |
| [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ vim desktop/conf/hue.ini  [sqoop]  # For autocompletion, fill out the librdbms section.  # Sqoop server URL  server\_url=http://hadoop102:12000/sqoop |

# 十一、Hue 与Spark 集成

## 1、安装jdk

### 1）解压

|  |
| --- |
| [alex@hadoop102 software]$ tar -xzvf jdk-8u144-linux-x64.tar.gz -C ../module/ |

### 2）配置环境变量

|  |
| --- |
| [alex@hadoop102 module]$ sudo vim /etc/profile  ##JAVA\_HOME  export JAVA\_HOME=/opt/module/jdk1.8.0\_144  export CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar  export PATH=$PATH:$JAVA\_HOME/bin  [alex@hadoop102 module]$ source /etc/profile  [alex@hadoop102 module]$ java -version |

## 2、安装Scala

### 1）解压

|  |
| --- |
| [alex@hadoop102 software]$ tar -xzvf scala-2.10.7.tgz -C /opt/module/ |

### 2）配置环境变量

|  |
| --- |
| [alex@hadoop102 module]$ sudo vim /etc/profile  ##SCALA\_HOME  export SCALA\_HOME=/opt/module/scala-2.10.7  export PATH=$PATH:$SCALA\_HOME/bin  [alex@hadoop102 module]$ source /etc/profile  [alex@hadoop102 module]$ scala -version  Scala code runner version 2.10.7 -- Copyright 2002-2017, LAMP/EPFL |

## 3、安装Maven

### 1）解压

|  |
| --- |
| [alex@hadoop102 software]$ tar -xzvf apache-maven-3.3.9-bin.tar.gz -C ../module/ |

### 2）配置环境变量

|  |
| --- |
| [alex@hadoop102 module]$ sudo vim /etc/profile  # MAVEN\_HOME  export MAVEN\_HOME=/opt/module/apache-maven-3.3.9  export PATH=$PATH:$MAVEN\_HOME/bin  export PATH=$PATH:$MAVEN\_HOME/conf  export MAVEN\_REPO=/opt/module/apache-maven-3.3.9/.m2/repository  [alex@hadoop102 module]$ source /etc/profile  [alex@hadoop102 module]$ mvn -v  Apache Maven 3.3.9 (bb52d8502b132ec0a5a3f4c09453c07478323dc5; 2015-11-11T00:41:47+08:00)  Maven home: /opt/module/apache-maven-3.3.9  Java version: 1.8.0\_144, vendor: Oracle Corporation  Java home: /opt/module/jdk1.8.0\_144/jre  Default locale: en\_US, platform encoding: UTF-8  OS name: "linux", version: "3.10.0-862.el7.x86\_64", arch: "amd64", family: "unix"  [alex@hadoop102 module]$ mvn -version  Apache Maven 3.3.9 (bb52d8502b132ec0a5a3f4c09453c07478323dc5; 2015-11-11T00:41:47+08:00)  Maven home: /opt/module/apache-maven-3.3.9  Java version: 1.8.0\_144, vendor: Oracle Corporation  Java home: /opt/module/jdk1.8.0\_144/jre  Default locale: en\_US, platform encoding: UTF-8  OS name: "linux", version: "3.10.0-862.el7.x86\_64", arch: "amd64", family: "unix" |

### 3）修改settings.xml配置文件

|  |
| --- |
| [alex@hadoop102 apache-maven-3.3.9]$ vim conf/settings.xml  <localRepository>/opt/module/apache-maven-3.3.9/m2/repository</localRepository>  <repositories>  <repository>  <id>cloudera</id>  <url>https://repository.cloudera.com/artifactory/cloudera-repos/</url>  <releases>  <enabled>true</enabled>  </releases>  <snapshots>  <enabled>false</enabled>  </snapshots>  </repository>  </repositories>  <mirror>  <id>alimaven</id>  <name>aliyun maven</name>  <url>http://maven.aliyun.com/nexus/content/repositories/central/</url>  <mirrorOf>central</mirrorOf>  </mirror>  <mirror>  <id>alimaven-new</id>  <mirrorOf>central</mirrorOf>  <name>aliyun maven</name>  <url>https://maven.aliyun.com/repository/central/</url>  </mirror>    <mirror>  <id>nexus-aliyun</id>  <name>Nexus aliyun</name>  <url>http://maven.aliyun.com/nexus/content/groups/public/</url>  <mirrorOf>central</mirrorOf>  </mirror>  <mirror>  <id>aliyun-repos</id>  <name>aliyun-repos</name>  <url>https://maven.aliyun.com/repository/public</url>  <mirrorOf>central</mirrorOf>  </mirror>  <mirror>  <id>sonatype-repos-s</id>  <name>sonatype-repos-s</name>  <url>https://oss.sonatype.org/content/repositories/snapshots</url>  <mirrorOf>central</mirrorOf>  </mirror>  <mirror>  <id>repo2</id>  <name>Mirror from Maven Repo2</name>  <url>http://repo2.maven.org/maven2/</url>  <mirrorOf>central</mirrorOf>  </mirror>  <mirror>  <id>centor</id>  <name>Mirror from Maven central</name>  <url>http://central.maven.org/maven2/</url>  <mirrorOf>central</mirrorOf>  </mirror>  <profile>  <id>jdk-1.8</id>  <activation>  <activeByDefault>true</activeByDefault>  <jdk>1.8</jdk>  </activation>  <properties>  <maven.compiler.source>1.8</maven.compiler.source>  <maven.compiler.target>1.8</maven.compiler.target>  <maven.compiler.compilerVersion>1.8</maven.compiler.compilerVersion>  </properties>  </profile> |

## 4、Spark编译源码

### 1）下载Spark

|  |
| --- |
| [alex@hadoop102 software]$wget \  http://archive.cloudera.com/cdh5/cdh/5/spark-1.2.0-cdh5.3.6-src.tar.gz |

### 2）解压

|  |
| --- |
| [alex@hadoop102 software]$ tar -xzvf cdh/spark-1.2.0-cdh5.3.6.tar.gz -C ../module/cdh/ |

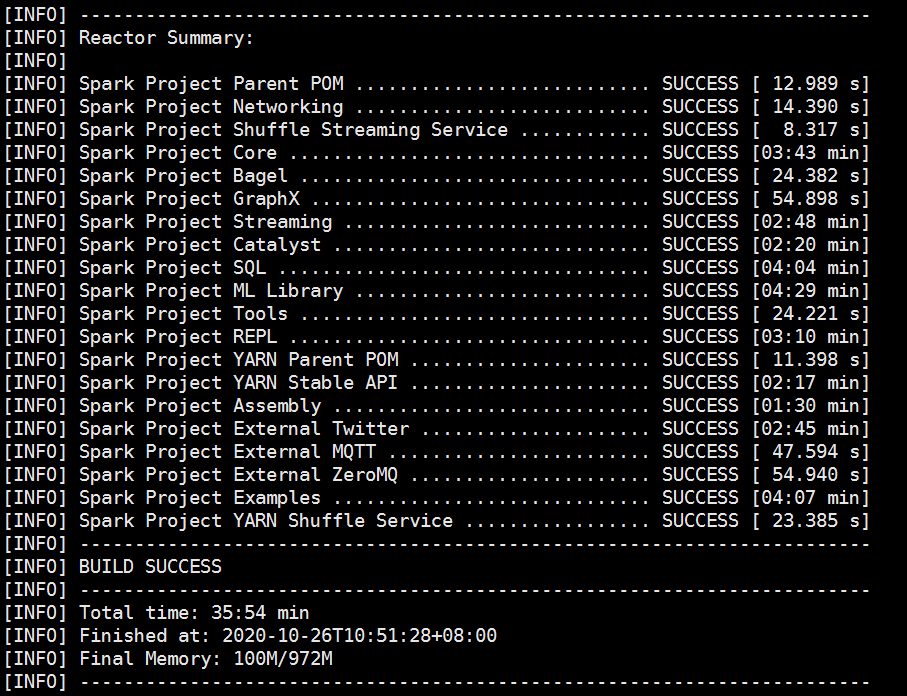
### 3）修改文件

|  |
| --- |
| [alex@hadoop102 spark-1.2.0-cdh5.3.6]$ vim make-distribution.sh  VERSION=1.2.0  SCALA\_VERSION=2.10.7  SPARK\_HADOOP\_VERSION=2.5.0-cdh5.3.6  SPARK\_HIVE=1  注释掉  # VERSION=$("$MVN" help:evaluate -Dexpression=project.version $@ 2>/dev/null | grep -v "INFO" | tail -n 1)  # SCALA\_VERSION=$("$MVN" help:evaluate -Dexpression=scala.binary.version $@ 2>/dev/null\  # | grep -v "INFO"\  # | tail -n 1)  #SPARK\_HADOOP\_VERSION=$("$MVN" help:evaluate -Dexpression=hadoop.version $@ 2>/dev/null\  # | grep -v "INFO"\  # | tail -n 1)  #SPARK\_HIVE=$("$MVN" help:evaluate -Dexpression=project.activeProfiles -pl sql/hive $@ 2>/dev/null\  # | grep -v "INFO"\  # | fgrep --count "<id>hive</id>";\  # Reset exit status to 0, otherwise the script stops here if the last grep finds nothing\  # because we use "set -o pipefail"  # echo -n) |

### 4）编译

|  |
| --- |
| [alex@hadoop102 spark-1.2.0-cdh5.3.6]$ ./make-distribution.sh \  --tgz \  -Phadoop-2.4 \  -Dhadoop.version=2.5.0-cdh5.3.6 \  -Pyarn \  -Phive-0.13.1 \  -Phive-thriftserver |

pom.xml



## 5、Spark安装部署

### 1）解压

|  |
| --- |
| [alex@hadoop102 cdh]$ tar -xzvf spark-1.2.0-bin-2.5.0-cdh5.3.6.tgz -C /opt/module/cdh/ |

### 2）配置spark-env.sh

|  |
| --- |
| [alex@hadoop102 module]$ cd cdh/spark-1.2.0-bin-2.5.0-cdh5.3.6/  [alex@hadoop102 spark-1.2.0-bin-2.5.0-cdh5.3.6]$ vim conf/spark-env.sh  #!/usr/bin/env bash  export SPARK\_MASTER\_IP=hadoop102  export MASTER=spark://hadoop102:7077  export SPARK\_WORKER\_MEMORY=1g  export JAVA\_HOME=/opt/module/jdk1.8.0\_144  export SCALA\_HOME=/opt/module/scala-2.10.7  SPARK\_LOG\_DIR=/opt/module/cdh/spark-1.2.0-cdh5.3.6/logs  spark.eventLog.dir=/user/alex/applicationHistory  spark.eventLog.enabled=true  spark.yarn.historyServer.address=http://hadoop102:19888  如果想 YARN ResourceManager 访问 Spark History Server ，则添加一行  spark.yarn.historyServer.address=http://HISTORY\_HOST:HISTORY\_PORT |

### 3）配置slaves

|  |
| --- |
| [alex@hadoop102 spark-1.2.0-bin-2.5.0-cdh5.3.6]$ vim conf/slaves  hadoop103  hadoop104 |

### 4）分发

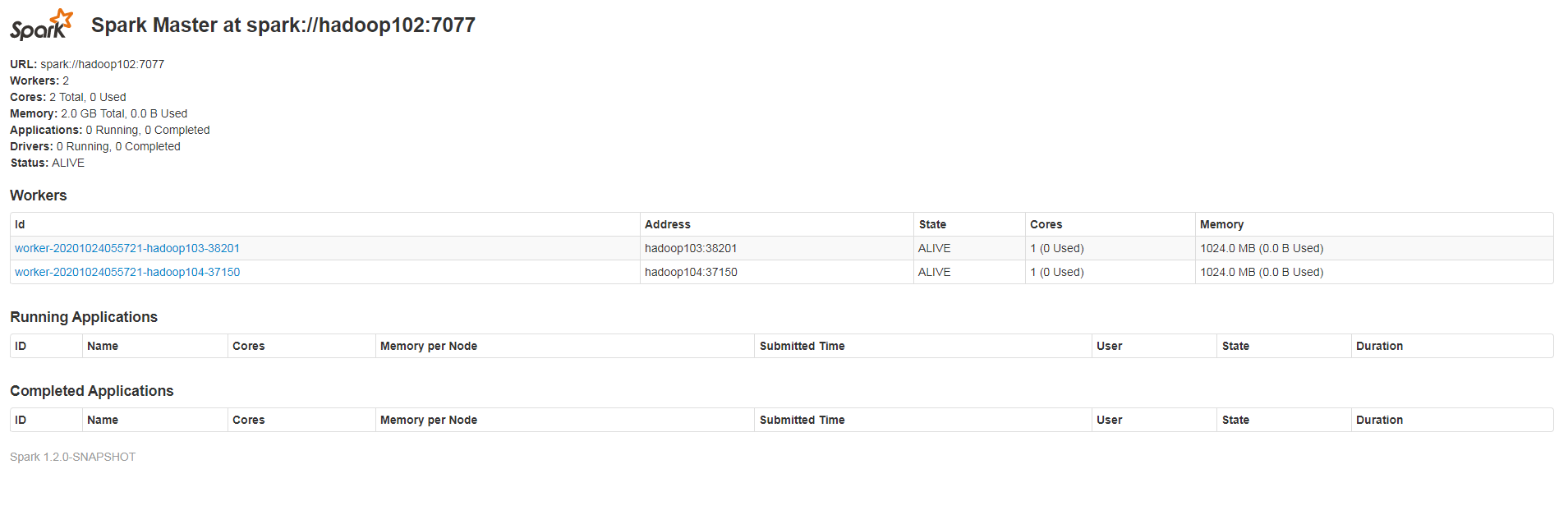
|  |
| --- |
| [alex@hadoop102 cdh]$ xsync spark-1.2.0-bin-2.5.0-cdh5.3.6 |

### 5）启动

|  |
| --- |
| [alex@hadoop102 spark-1.2.0-bin-2.5.0-cdh5.3.6]$ sbin/start-all.sh |

### 6）浏览器访问

|  |
| --- |
| http://192.168.2.102:8080/ |



## 6、与Hue整合

### 1）编辑core-site.xml

|  |
| --- |
| [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ vim etc/hadoop/core-site.xml  <!-- spark 用户代理 -->  <property>  <name>hadoop.proxyuser.sparkuser.hosts</name>  <value>\*</value>  </property>  <property>  <name>hadoop.proxyuser.sparkuser.groups</name>  <value>\*</value>  </property> |

### 2）创建hdfs目录

|  |
| --- |
| [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ hadoop fs -mkdir -P /user/spark/applicationHistory  [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ hadoop fs -mkdir -chmod -R 755 /user/spark |

### 3）修改hue.ini配置文件

|  |
| --- |
| [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ vim desktop/conf/hue.ini  [spark]  # URL of the REST Spark Job Server.  server\_url=http://hadoop102:8090/ |

### 4）启动Spark

|  |
| --- |
| [alex@hadoop102 spark-2.1.1-bin-hadoop2.7]$ sbin/start-all.sh  [alex@hadoop102 spark-2.1.1-bin-hadoop2.7]$ sbin/start-history-server.sh  [alex@hadoop102 spark-2.1.1-bin-hadoop2.7]$ sbin/start-thriftserver.sh \  --master spark://hadoop102:7077 --deploy-mode client |

### 5）重启HUE测试

|  |
| --- |
| [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ build/env/bin/supervisor |

# 十二、总结

在此我们总结一下集成HUE时，我们开启的后台服务项

## 1、Hadoop

|  |
| --- |
| [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ sbin/httpfs.sh start  [alex@hadoop102 hadoop-2.5.0-cdh5.3.6]$ sbin/httpfs.sh stop |

## 2、Hive

|  |
| --- |
| [alex@hadoop102 hive-0.13.1-cdh5.3.6]$ bin/hive --service metastore &  [alex@hadoop102 hive-0.13.1-cdh5.3.6]$ bin/hive --service hiveserver2 & |

## 3、Hbase

|  |
| --- |
| [alex@hadoop102 hbase-0.98.6-cdh5.3.6]$ bin/start-hbase.sh  [alex@hadoop102 hbase-0.98.6-cdh5.3.6]$ bin/stop-hbase.sh  [alex@hadoop102 hbase-0.98.6-cdh5.3.6]$ bin/hbase-daemon.sh start thrift  [alex@hadoop102 hbase-0.98.6-cdh5.3.6]$ bin/hbase-daemon.sh stop thrift |

## 4、Oozie

|  |
| --- |
| [alex@hadoop102 oozie-4.0.0-cdh5.3.6]$ bin/oozied.sh start |

## 5、Spark

|  |
| --- |
| [alex@hadoop102 spark-2.1.1-bin-hadoop2.7]$ sbin/start-all.sh  [alex@hadoop102 spark-2.1.1-bin-hadoop2.7]$ sbin/stop-all.sh  [alex@hadoop102 spark-2.1.1-bin-hadoop2.7]$ sbin/start-history-server.sh  [alex@hadoop102 spark-2.1.1-bin-hadoop2.7]$ sbin/stop-history-server.sh  [alex@hadoop102 spark-2.1.1-bin-hadoop2.7]$ sbin/start-thriftserver.sh \  --master spark://hadoop102:7077 --deploy-mode client  [alex@hadoop102 spark-2.1.1-bin-hadoop2.7]$sbin/stop-thriftserver.sh \  --master spark://hadoop102:7077 --deploy-mode client |

# 十三、pseudo-distributed.ini

|  |
| --- |
| [desktop]  secret\_key=jFE93j;2[290-eiw.KEiwN2s3['d;/.q[eIW^y#e=+Iei\*@Mn<qW5o  http\_host=hadoop102  http\_port=8888  time\_zone=Asia/Shanghai  [notebook]  [[interpreters]]  [[[hive]]]  name=Hive  interface=hiveserver2  [[[impala]]]  name=Impala  interface=hiveserver2  [[[sparksql]]]  name=SparkSql  interface=hiveserver2  [[[spark]]]  name=Scala  interface=livy  [[[pyspark]]]  name=PySpark  interface=livy  [[[r]]]  name=R  interface=livy  [[[jar]]]  name=Spark Submit Jar  interface=livy-batch  [hadoop]  [[hdfs\_clusters]]  [[[default]]]  fs\_defaultfs=hdfs://hadoop102:9000  webhdfs\_url=http://hadoop102:14000/webhdfs/v1  hadoop\_conf\_dir=/opt/module/cdh/hadoop-2.5.0-cdh5.3.6/etc/hadoop  hadoop\_hdfs\_home=/opt/module/cdh/hadoop-2.5.0-cdh5.3.6 hadoop\_bin=/opt/module/cdh/hadoop-2.5.0-cdh5.3.6/bin  [[yarn\_clusters]]  [[[default]]]  resourcemanager\_host=hadoop103  resourcemanager\_port=8032  submit\_to=True  resourcemanager\_api\_url=http://hadoop103:8088  proxy\_api\_url=http://hadoop103:8088  history\_server\_api\_url=http://hadoop102:19888  [beeswax]  hive\_server\_host=hadoop102  hive\_server\_port=10000  hive\_conf\_dir=/opt/module/cdh/hive-0.13.1-cdh5.3.6/conf  [librdbms]  [[databases]]  [[[mysql]]]  nice\_name="ZMysqlDB"  engine=mysql  host=hadoop102  port=3306  user=root  password=000000  [hbase]  hbase\_clusters=(Cluster|hadoop102:9090)  hbase\_conf\_dir=/opt/module/cdh/hbase-0.98.6-cdh5.3.6/conf  [zookeeper]  [[clusters]]  [[[default]]]  host\_ports=hadoop102:2181,hadoop103:2181,hadoop104:2181  [spark]  livy\_server\_host=hadoop102  livy\_server\_port=8998  livy\_server\_session\_kind=yarn  security\_enabled=false  sql\_server\_host=hadoop102  sql\_server\_port=10000 |

# 十四、CDH 的 Yum源

|  |
| --- |
| [root@hadoop102 ~]# cd /etc/yum.repos.d/  [root@hadoop102 ~]# touch cloudera.repo  [root@hadoop102 ~]# vim cloudera.repo  [cloudera-cdh5]  name=Cloudera's Distribution for Hadoop, Version 5  baseurl= http://archive.cloudera.com/cdh5/redhat/6/x86\_64/cdh/5/  gpgkey = http://archive.cloudera.com/cdh5/redhat/6/x86\_64/cdh/RPM-GPG-KEY-cloudera  gpgcheck = 1    [cloudera-cm5]  name=Cloudera's Distribution for Hadoop, Version 5  baseurl= http://archive.cloudera.com/cm5/redhat/6/x86\_64/cm/5/  gpgkey = http://archive.cloudera.com/cm5/redhat/6/x86\_64/cm/RPM-GPG-KEY-cloudera  gpgcheck = 1 |

wget http://archive.cloudera.com/cdh5/redhat/6/x86\_64/cdh/cloudera-cdh5.repo

mv cloudera-cdh5.repo /ect/yum.repo.d/

# 十五、Apache Hue 安装部署

## 1、解压

|  |
| --- |
| [alex@hadoop102 software]$ tar -xzvf hue-4.1.0.tgz -C ../module/ |

## 2、编译

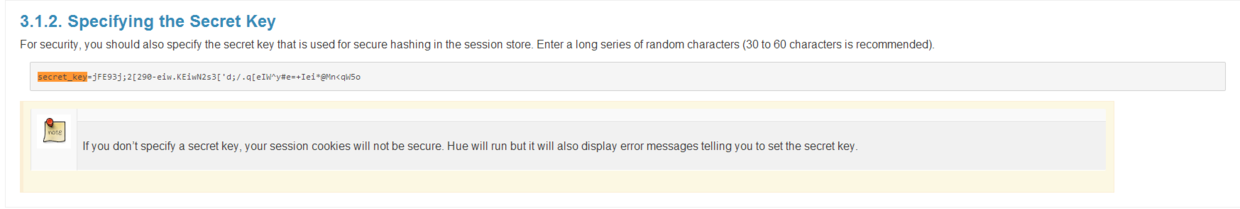
|  |
| --- |
| [alex@hadoop102 hue-4.1.0]$ make apps |

## 3、配置HUE

修改Hue.ini文件

文件位置：/opt/module/hue-4.1.0/desktop/conf/hue.ini

其中的secret\_key请参照官方网站配置：



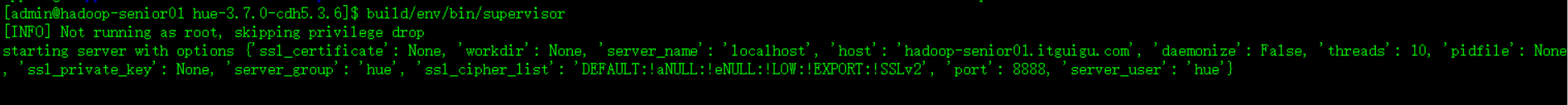
|  |
| --- |
| [alex@hadoop102 hue-4.1.0]$ vim desktop/conf/hue.ini  secret\_key=jFE93j;2[290-eiw.KEiwN2s3['d;/.q[eIW^y#e=+Iei\*@Mn<qW5o  http\_host=hadoop102  http\_port=8888  time\_zone=Asia/Shanghai |

## 4、启动Hue

完成之后呢，保存退出，我们来使用命令启动Hue

|  |
| --- |
| [alex@hadoop102 hue-4.1.0]$ build/env/bin/supervisor |

出现如下界面表示启动成功

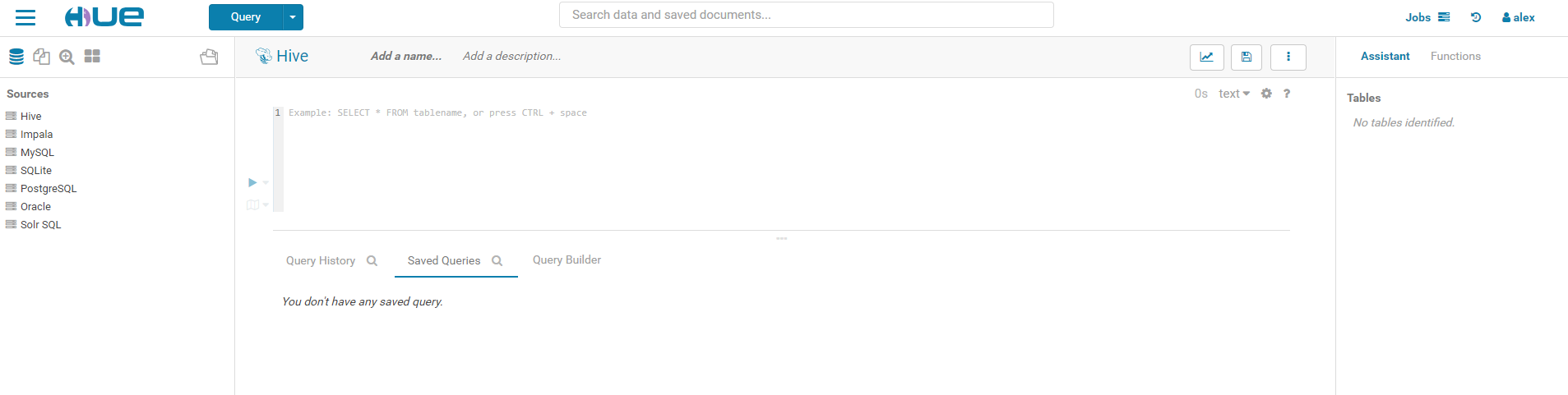


## 5、浏览器访问

|  |
| --- |
| http://hadoop102:8888 |

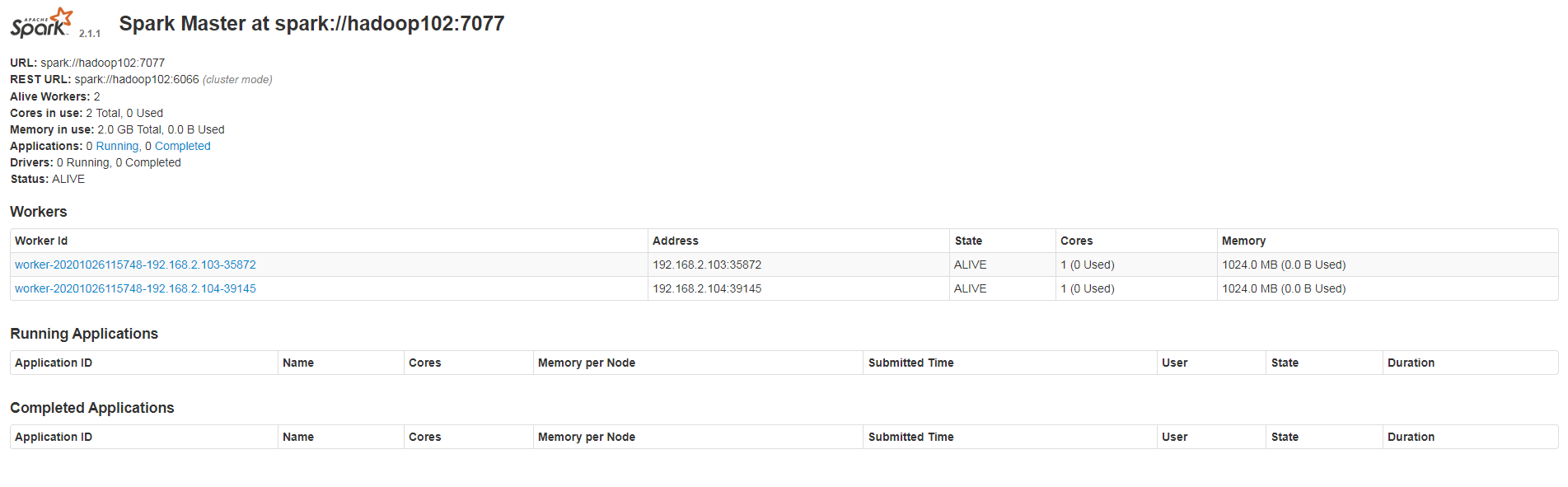


这句话是在提示你，第一次使用本工具，需要创建一个用户及密码，且会成为hue的超级用户凭证，在此呢，我设置为alex用户名，密码随意，那就000000吧，然后呢就可以见到如下界面了



配置hue.ini

|  |
| --- |
| [alex@hadoop102 hue-3.7.0-cdh5.3.6]$ vim desktop/conf/hue.ini  [notebook]  [[interpreters]]  [[[hive]]]  name=Hive  interface=hiveserver2  [[[impala]]]  name=Impala  interface=hiveserver2  [[[sparksql]]]  name=SparkSql  interface=hiveserver2  [[[spark]]]  name=Scala  interface=livy  [[[pyspark]]]  name=PySpark  interface=livy  [[[r]]]  name=R  interface=livy  [[[jar]]]  name=Spark Submit Jar  interface=livy-batch  [spark]  livy\_server\_host=hadoop102  livy\_server\_port=8998  livy\_server\_session\_kind=yarn  security\_enabled=false  sql\_server\_host=hadoop102  sql\_server\_port=10000 |



# 十六、Apache Spark 安装部署

[ Standalone模式]

## 1、解压

|  |
| --- |
| [alex@hadoop102 software]$ tar -xzvf spark-2.1.1-bin-hadoop2.7.tgz -C /opt/module |

## 2、配置环境

|  |
| --- |
| [alex@hadoop102 module]$ sudo vim /etc/profile  #spark path  export SPARK\_HOME=/opt/module/spark-2.1.1-bin-hadoop2.7  [alex@hadoop102 module]$ source /etc/profile |

## 3、编辑slave文件

将work的hostname写入

|  |
| --- |
| [alex@hadoop102 spark-2.1.1-bin-hadoop2.7]$ vim conf/slaves  hadoop103  hadoop104 |

## 4、编辑spark-env.sh

|  |
| --- |
| [alex@hadoop102 spark-2.1.1-bin-hadoop2.7]$ vim conf/spark-env.sh  SPARK\_MASTER\_HOST=hadoop102  SPARK\_MASTER\_PORT=7077  JAVA\_HOME=/opt/module/jdk1.8.0\_144  SCALA\_HOME=/opt/module/scala-2.11.8  SPARK\_LOG\_DIR=/opt/module/spark-2.1.1-bin-hadoop2.7/log |

## 5、修改JAVA\_HOME

|  |
| --- |
| [alex@hadoop102 spark-2.1.1-bin-hadoop2.7]$ vim sbin/spark-config.sh  # JAVA\_HOME  export JAVA\_HOME=/opt/module/jdk1.8.0\_144  # SCALA\_HOME  export SCALA\_HOME=/opt/module/scala-2.11.8 |

## 6、分发

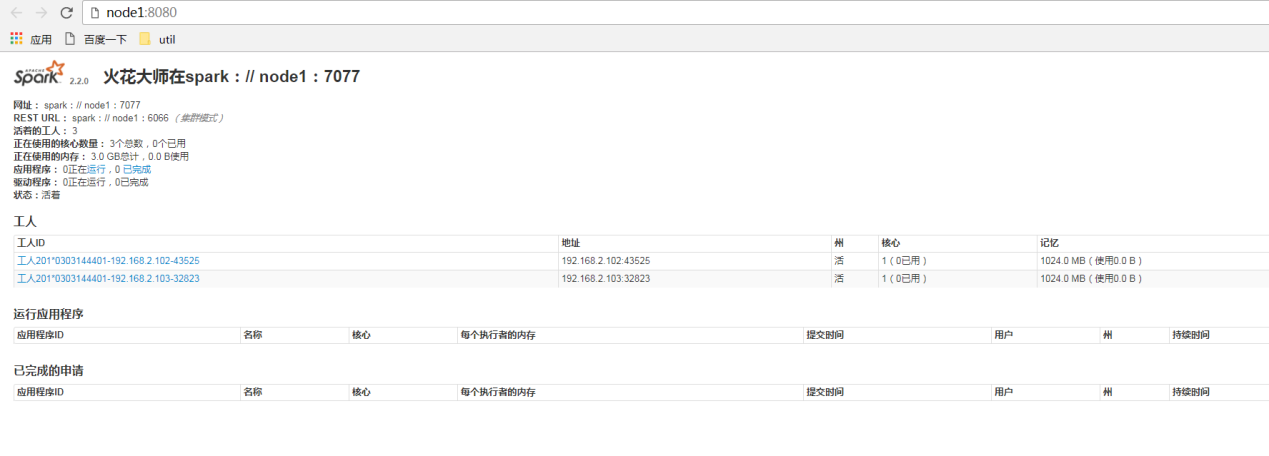
|  |
| --- |
| [alex@hadoop102 module]$ xsync spark-2.1.1-bin-hadoop2.7 |

## 7、启动

|  |
| --- |
| [alex@hadoop102 spark-2.1.1-bin-hadoop2.7]$ sbin/start-all.sh |

## 8、wen浏览器访问

|  |
| --- |
| http://hadoop102:8080/ |

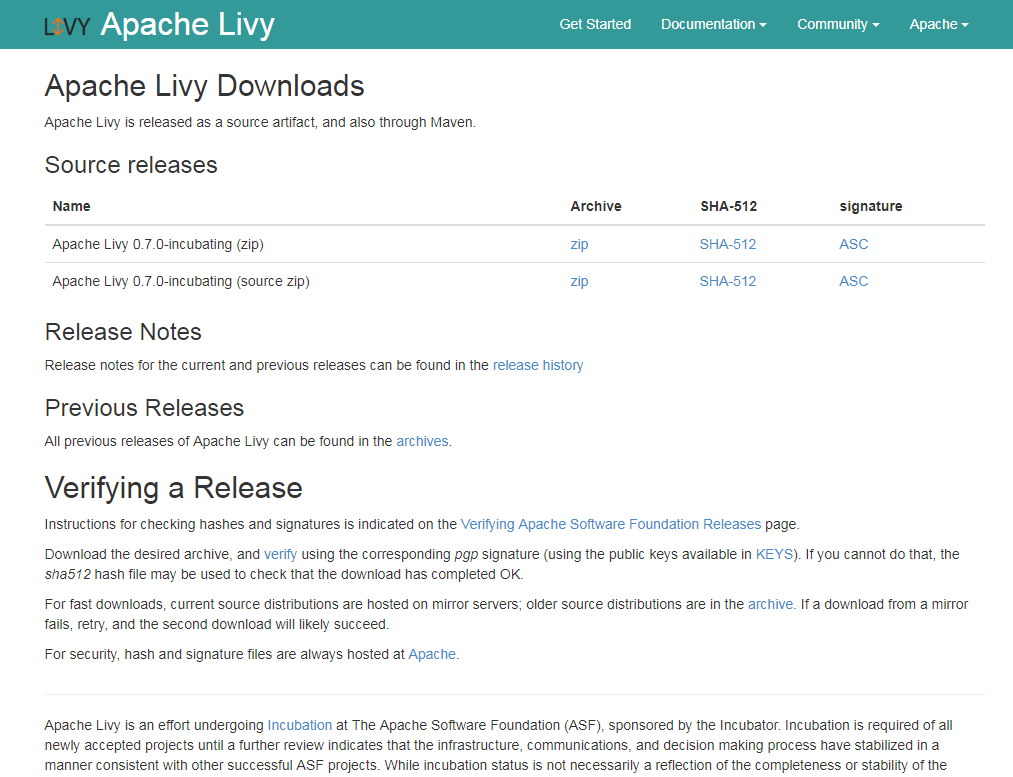


# 十七、Apache Hue 与 Apache Spark 集成

Spark 是 Standalone模式部署的

## 1、下载

http://livy.incubator.apache.org/download/



## 2、解压

|  |
| --- |
| [alex@hadoop102 software]$ unzip livy-0.5.0-incubating-bin.zip |

## 3、移动并修改名字

|  |
| --- |
| [alex@hadoop102 software]$ mv livy-0.5.0-incubating-bin ../module/livy-0.5.0 |

## 4、修改livy.conf

|  |
| --- |
| [alex@hadoop102 software]$ cd ../module/livy-0.5.0/  [alex@hadoop102 livy-0.5.0]$ vim conf/livy.conf  livy.server.host = hadoop102  livy.server.port = 8998  livy.spark.master = spark://hadoop102:7077  livy.spark.deploy-mode = client |

## 5、修改livy-env.sh

|  |
| --- |
| [alex@hadoop102 livy-0.5.0]$ vim conf/livy-env.sh  export JAVA\_HOME=/opt/module/jdk1.8.0\_144  export SPARK\_HOME=/opt/module/spark-2.1.1-bin-hadoop2.7  export SPARK\_CONF\_DIR=/opt/module/spark-2.1.1-bin-hadoop2.7/conf  export LIVY\_HOME=/opt/module/livy-0.5.0  export LIVY\_LOG\_DIR=/opt/module/livy-0.5.0/logs |

## 6、创建logs文件夹

|  |
| --- |
| [alex@hadoop102 livy-0.5.0]$ mkdir logs |

## 7、启动 livy Server

|  |
| --- |
| [alex@hadoop102 livy-0.5.0]$ bin/livy-server |

## 8、启动spark thriftserver

|  |
| --- |
| [alex@hadoop102 spark-2.1.1-bin-hadoop2.7]$ sbin/start-thriftserver.sh \  --master spark://hadoop102:7077 --deploy-mode client |

## 9、启动Spark

|  |
| --- |
| [alex@hadoop102 spark-2.1.1-bin-hadoop2.7]$ sbin/start-all.sh |

## 10、启动Hue

|  |
| --- |
| [alex@hadoop102 hue-4.1.0]$ build/env/bin/supervisor |

## 11、测试

print(1)

