**Apache Hue**

# 课程计划

目录

[一、 课程计划 2](#_Toc11055136)

[二、 Apache Hue介绍 4](#_Toc11055137)

[1． Hue是什么 4](#_Toc11055138)

[2． Hue能做什么 5](#_Toc11055139)

[3． Hue的架构 6](#_Toc11055140)

[三、 Hue的安装 7](#_Toc11055141)

[1． 上传解压安装包 7](#_Toc11055142)

[2． 编译初始化工作 7](#_Toc11055143)

[2.1． 联网安装各种必须的依赖包 7](#_Toc11055144)

[2.2． Hue初始化配置 8](#_Toc11055145)

[2.3． 创建mysql中Hue使用的DB 8](#_Toc11055146)

[3． 编译Hue 9](#_Toc11055147)

[4． 启动Hue、Web UI访问 10](#_Toc11055148)

[四、 Hue与软件的集成 11](#_Toc11055149)

[1． Hue集成HDFS 11](#_Toc11055150)

[1.1． 修改core-site.xml配置 11](#_Toc11055151)

[1.2． 修改hdfs-site.xml配置 11](#_Toc11055152)

[1.3． 修改hue.ini 12](#_Toc11055153)

[1.4． 重启HDFS、Hue 12](#_Toc11055154)

[2． Hue集成YARN 13](#_Toc11055155)

[2.1． 修改hue.ini 13](#_Toc11055156)

[2.2． 开启yarn日志聚集服务 13](#_Toc11055157)

[2.3． 重启Yarn、Hue 13](#_Toc11055158)

[3． Hue集成Hive 14](#_Toc11055159)

[3.1． 修改Hue.ini 14](#_Toc11055160)

[3.2． 启动Hive服务、重启hue 14](#_Toc11055161)

[4． Hue集成Mysql 16](#_Toc11055162)

[4.1． 修改hue.ini 16](#_Toc11055163)

[4.2． 重启hue 16](#_Toc11055164)

[5． Hue集成Oozie 17](#_Toc11055165)

[5.1． 修改hue配置文件hue.ini 17](#_Toc11055166)

[5.2． 启动hue、oozie 18](#_Toc11055167)

[5.3． 使用hue配置oozie调度 19](#_Toc11055168)

[5.4． 利用hue调度shell脚本 19](#_Toc11055169)

[5.5． 利用hue调度hive脚本 22](#_Toc11055170)

[5.6． 利用hue调度MapReduce程序 24](#_Toc11055171)

[5.7． 利用Hue配置定时调度任务 25](#_Toc11055172)

[6． Hue集成Hbase 27](#_Toc11055173)

[6.1． 修改hbase配置 27](#_Toc11055174)

[6.2． 修改hadoop配置 27](#_Toc11055175)

[6.3． 修改Hue配置 28](#_Toc11055176)

[6.4． 启动hbase(包括thrift服务)、hue 28](#_Toc11055177)

[7． Hue集成Impala 30](#_Toc11055178)

[7.1． 修改Hue.ini 30](#_Toc11055179)

[7.2． 重启Hue 30](#_Toc11055180)

# Apache Hue介绍

## Hue是什么

HUE=Hadoop User Experience

Hue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，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，例如操作HDFS上的数据，运行MapReduce Job，执行Hive的SQL语句，浏览HBase数据库等等。



## Hue能做什么

访问HDFS和文件浏览

通过web调试和开发hive以及数据结果展示

查询solr和结果展示，报表生成

通过web调试和开发impala交互式SQL Query

spark调试和开发

Pig开发和调试

oozie任务的开发，监控，和工作流协调调度

Hbase数据查询和修改，数据展示

Hive的元数据（metastore）查询

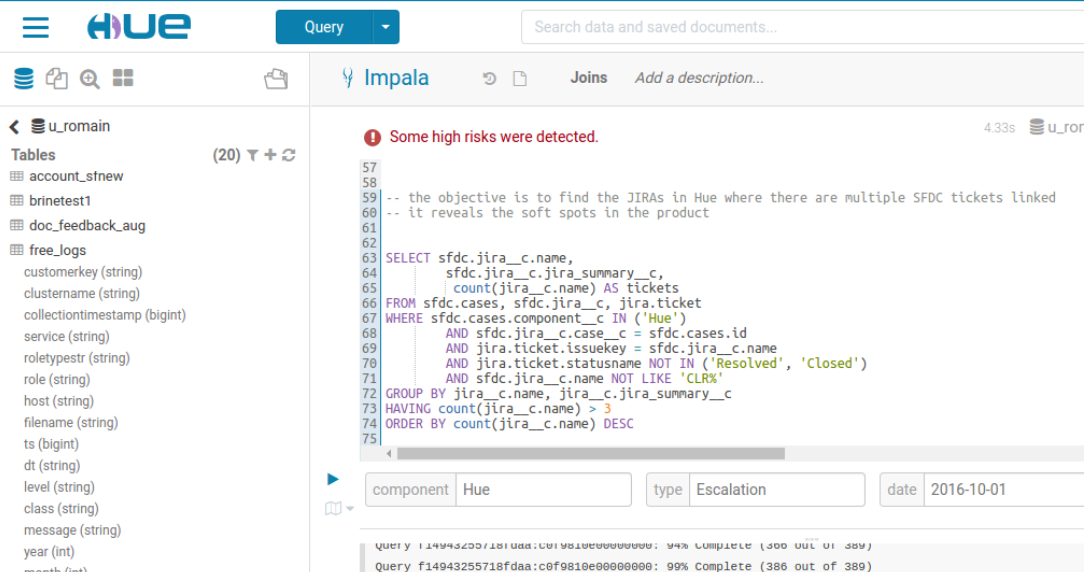
MapReduce任务进度查看，日志追踪

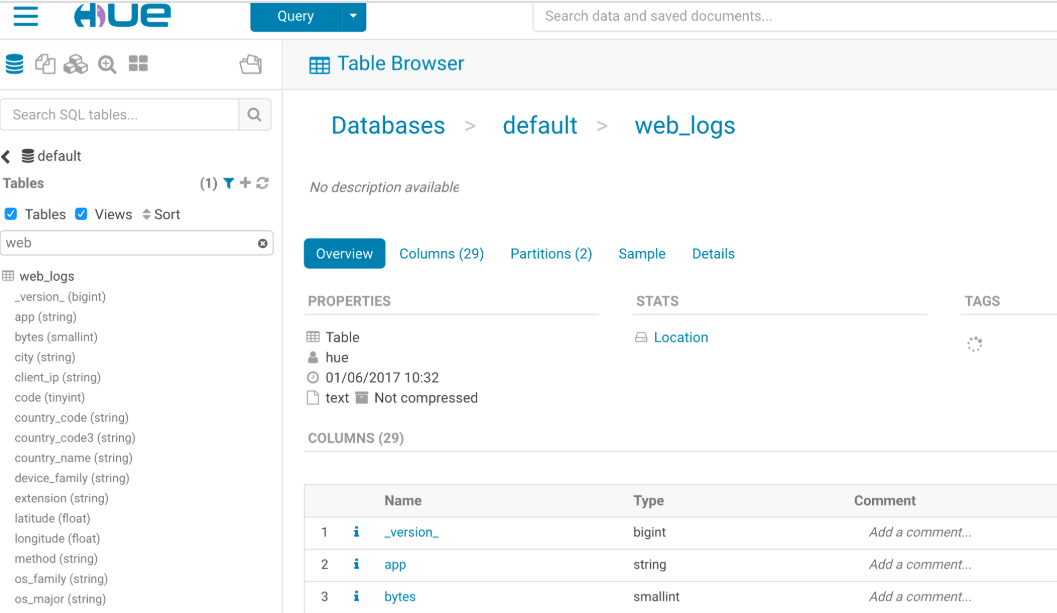
创建和提交MapReduce，Streaming，Java job任务

Sqoop2的开发和调试

Zookeeper的浏览和编辑

数据库（MySQL，PostGres，SQlite，Oracle）的查询和展示

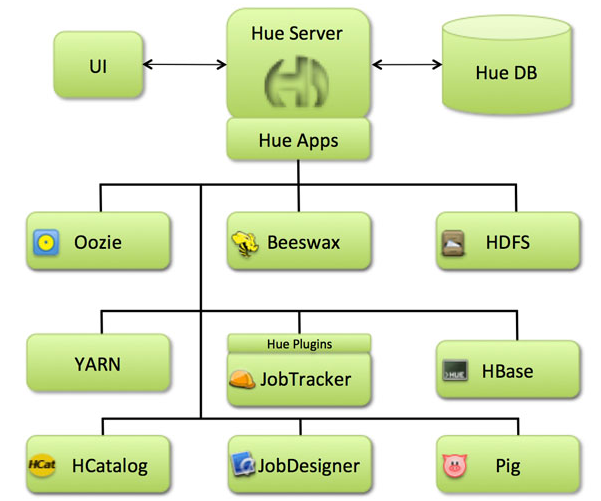




## Hue的架构

Hue是一个友好的界面集成框架，可以集成各种大量的大数据体系软件框架，通过一个界面就可以做到查看以及执行所有的框架。

Hue提供的这些功能相比Hadoop生态各组件提供的界面更加友好，但是一些需要debug的场景可能还是要使用原生系统才能更加深入的找到错误的原因。



# Hue的安装

## 上传解压安装包

Hue的安装支持多种方式，包括rpm包的方式进行安装、tar.gz包的方式进行安装以及cloudera manager的方式来进行安装等，我们这里使用tar.gz包的方式来进行安装。

Hue的压缩包的下载地址：

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

我们这里使用的是CDH5.14.0这个对应的版本，具体下载地址为

<http://archive.cloudera.com/cdh5/cdh/5/hue-3.9.0-cdh5.14.0.tar.gz>

cd /export/servers/

tar -zxvf hue-3.9.0-cdh5.14.0.tar.gz

## 编译初始化工作

### 联网安装各种必须的依赖包

|  |
| --- |
| yum install -y asciidoc cyrus-sasl-devel cyrus-sasl-gssapi cyrus-sasl-plain gcc gcc-c++ krb5-devel libffi-devel libxml2-devel libxslt-devel make openldap-devel python-devel sqlite-devel gmp-devel |

### Hue初始化配置

cd /export/servers/hue-3.9.0-cdh5.14.0/desktop/conf

vim hue.ini

|  |
| --- |
| #通用配置  [desktop]  secret\_key=jFE93j;2[290-eiw.KEiwN2s3['d;/.q[eIW^y#e=+Iei\*@Mn<qW5o  http\_host=node-1  is\_hue\_4=true  time\_zone=Asia/Shanghai  server\_user=root  server\_group=root  default\_user=root  default\_hdfs\_superuser=root  #配置使用mysql作为hue的存储数据库,大概在hue.ini的587行左右  [[database]]  engine=mysql  host=node-1  port=3306  user=root  password=Hadoop  **name=hue** |

### 创建mysql中Hue使用的DB

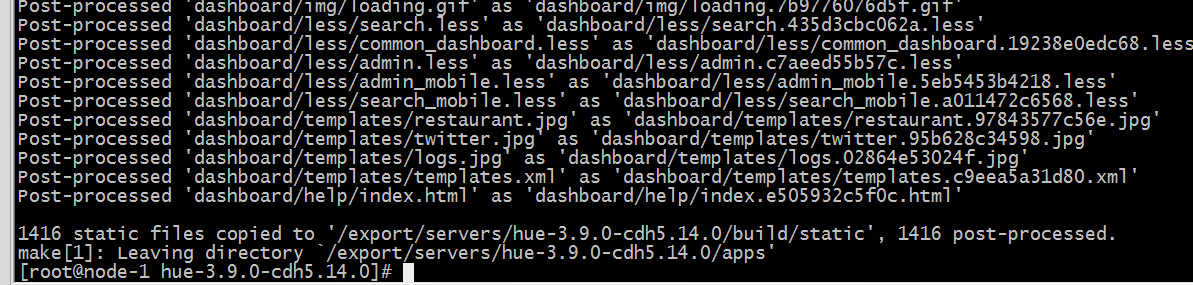
create database hue default character set utf8 default collate utf8\_general\_ci;

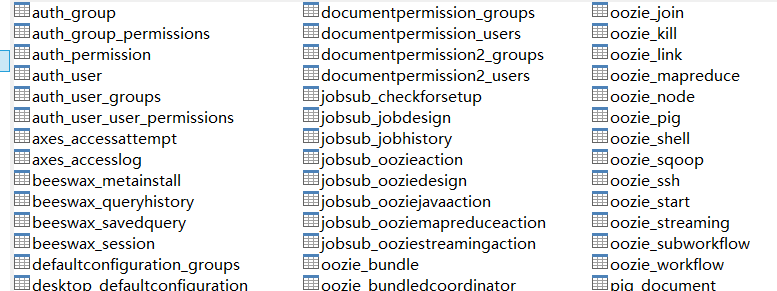
## 编译Hue

cd /export/servers/hue-3.9.0-cdh5.14.0

make apps

编译成功之后，会在hue数据库中创建许多初始化表。





## 启动Hue、Web UI访问

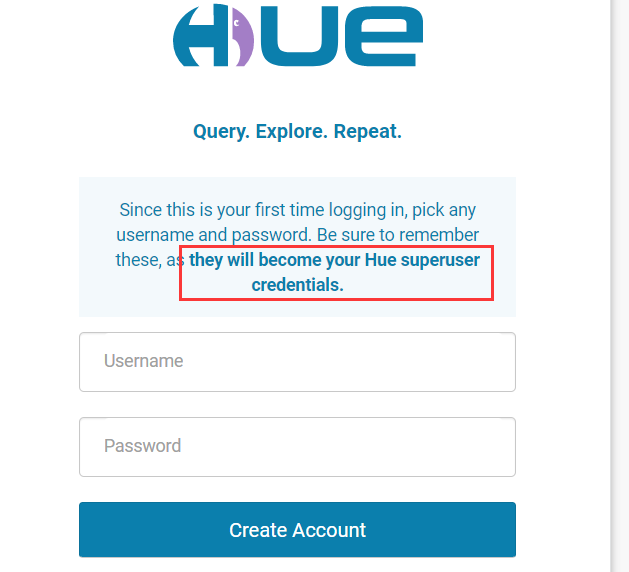
cd /export/servers/hue-3.9.0-cdh5.14.0/

build/env/bin/supervisor

页面访问路径：

<http://node-1:8888>

第一次访问的时候，**需要设置超级管理员用户和密码。记住它。**



若想关闭Hue ,直接在窗口ctrl+c即可。

# Hue与软件的集成

## Hue集成HDFS

注意修改完HDFS相关配置后，需要把配置scp给集群中每台机器，重启hdfs集群。

### 修改core-site.xml配置

|  |
| --- |
| <!—允许通过httpfs方式访问hdfs的主机名 -->  <property>  <name>hadoop.proxyuser.root.hosts</name>  <value>\*</value>  </property>  <!—允许通过httpfs方式访问hdfs的用户组 -->  <property>  <name>hadoop.proxyuser.root.groups</name>  <value>\*</value>  </property> |

### 修改hdfs-site.xml配置

|  |
| --- |
| <property>  <name>dfs.webhdfs.enabled</name>  <value>true</value>  </property> |

### 修改hue.ini

cd /export/servers/hue-3.9.0-cdh5.14.0/desktop/conf

vim hue.ini

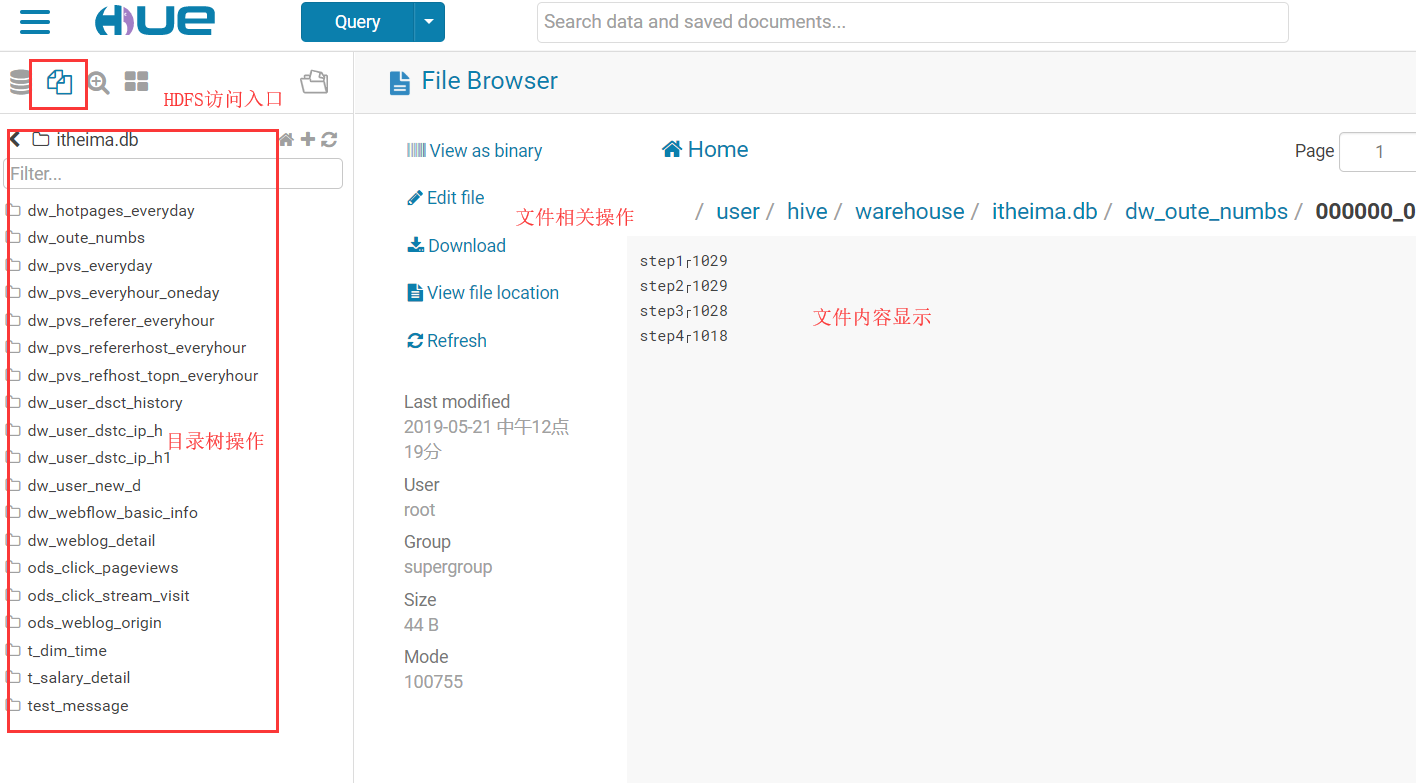
|  |
| --- |
| [[hdfs\_clusters]]  [[[default]]]  fs\_defaultfs=hdfs://node-1:9000  webhdfs\_url=http://node-1:50070/webhdfs/v1  hadoop\_hdfs\_home= /export/servers/hadoop-2.7.5  hadoop\_bin=/export/servers/hadoop-2.7.5/bin  hadoop\_conf\_dir=/export/servers/hadoop-2.7.5/etc/hadoop |

### 重启HDFS、Hue

start-dfs.sh

cd /export/servers/hue-3.9.0-cdh5.14.0/

build/env/bin/supervisor



## Hue集成YARN

### 修改hue.ini

|  |
| --- |
| [[yarn\_clusters]]  [[[default]]]  resourcemanager\_host=node-1  resourcemanager\_port=8032  submit\_to=True  resourcemanager\_api\_url=http://node-1:8088  history\_server\_api\_url=http://node-1:19888 |

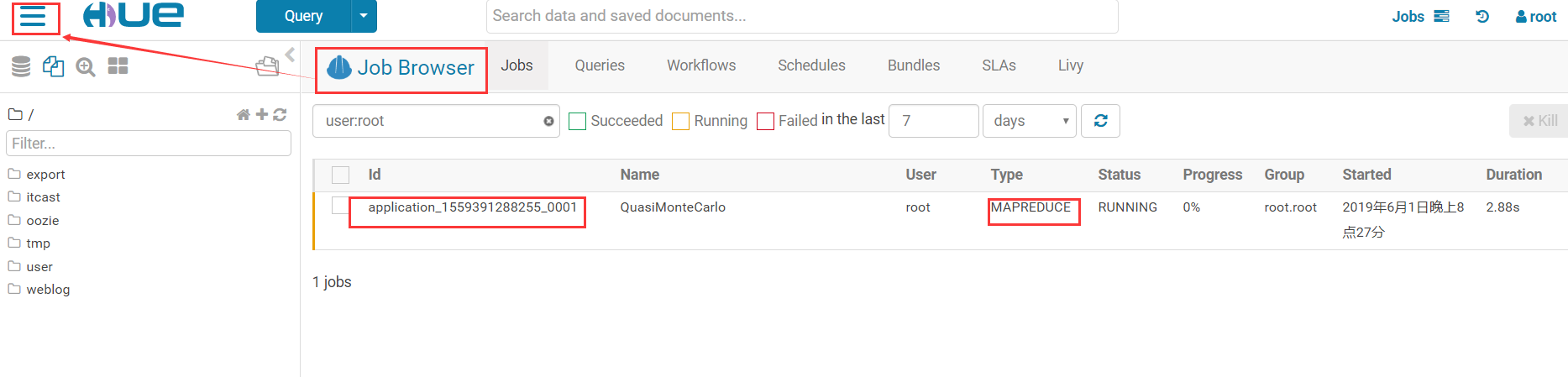
### 开启yarn日志聚集服务

MapReduce 是在各个机器上运行的， 在运行过程中产生的日志存在于各个机器上，为了能够统一查看各个机器的运行日志，将日志集中存放在 HDFS 上， 这个过程就是日志聚集。

|  |
| --- |
| <property> ##是否启用日志聚集功能。  <name>yarn.log-aggregation-enable</name>  <value>true</value>  </property>  <property> ##设置日志保留时间，单位是秒。  <name>yarn.log-aggregation.retain-seconds</name>  <value>106800</value>  </property> |

### 重启Yarn、Hue

build/env/bin/supervisor



## Hue集成Hive

如果需要配置hue与hive的集成，我们需要启动hive的metastore服务以及hiveserver2服务（impala需要hive的metastore服务，hue需要hvie的hiveserver2服务）。

### 修改Hue.ini

|  |
| --- |
| [beeswax]  hive\_server\_host=node-1  hive\_server\_port=10000  hive\_conf\_dir=/export/servers/hive/conf  server\_conn\_timeout=120  auth\_username=root  auth\_password=123456  [metastore]  #允许使用hive创建数据库表等操作  enable\_new\_create\_table=true |

### 启动Hive服务、重启hue

去node-1机器上启动hive的metastore以及hiveserver2服务

cd /export/servers/hive

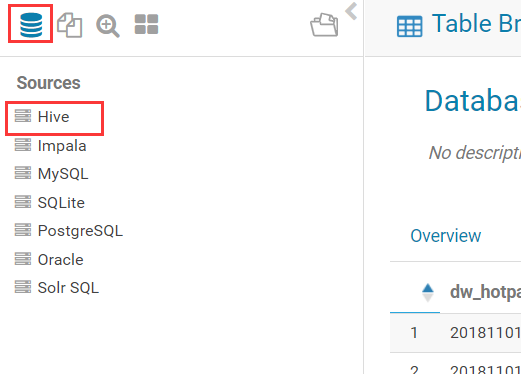
nohup bin/hive --service metastore &

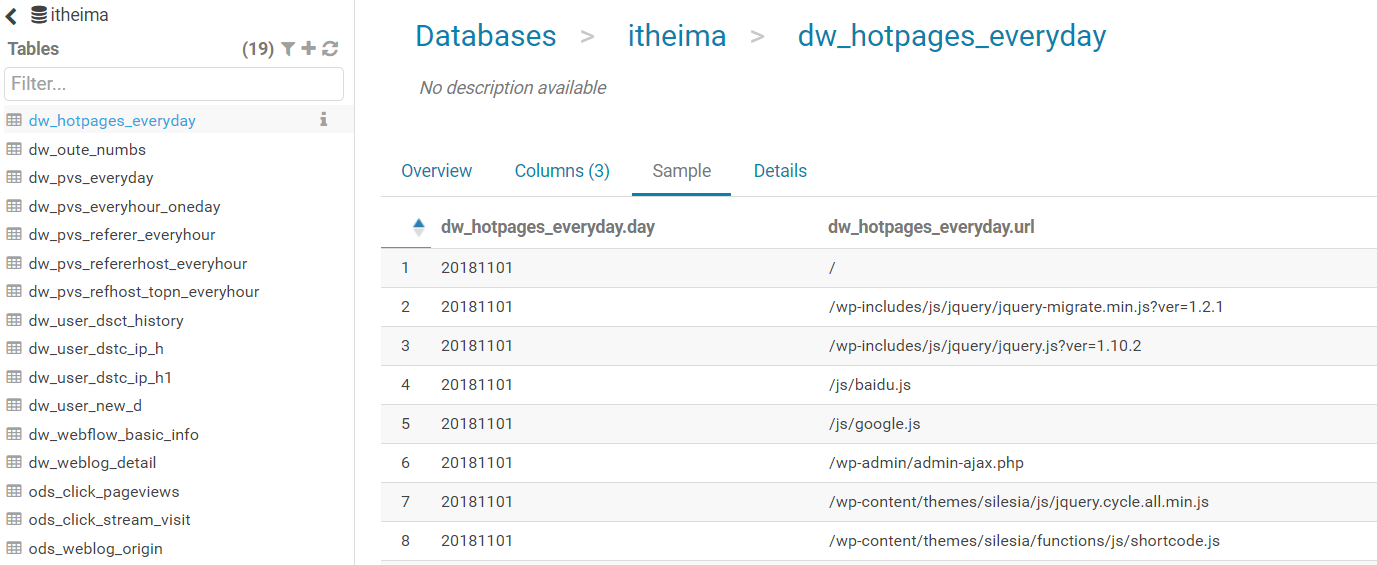
nohup bin/hive --service hiveserver2 &

重新启动hue。

cd /export/servers/hue-3.9.0-cdh5.14.0/

build/env/bin/supervisor





## Hue集成Mysql

### 修改hue.ini

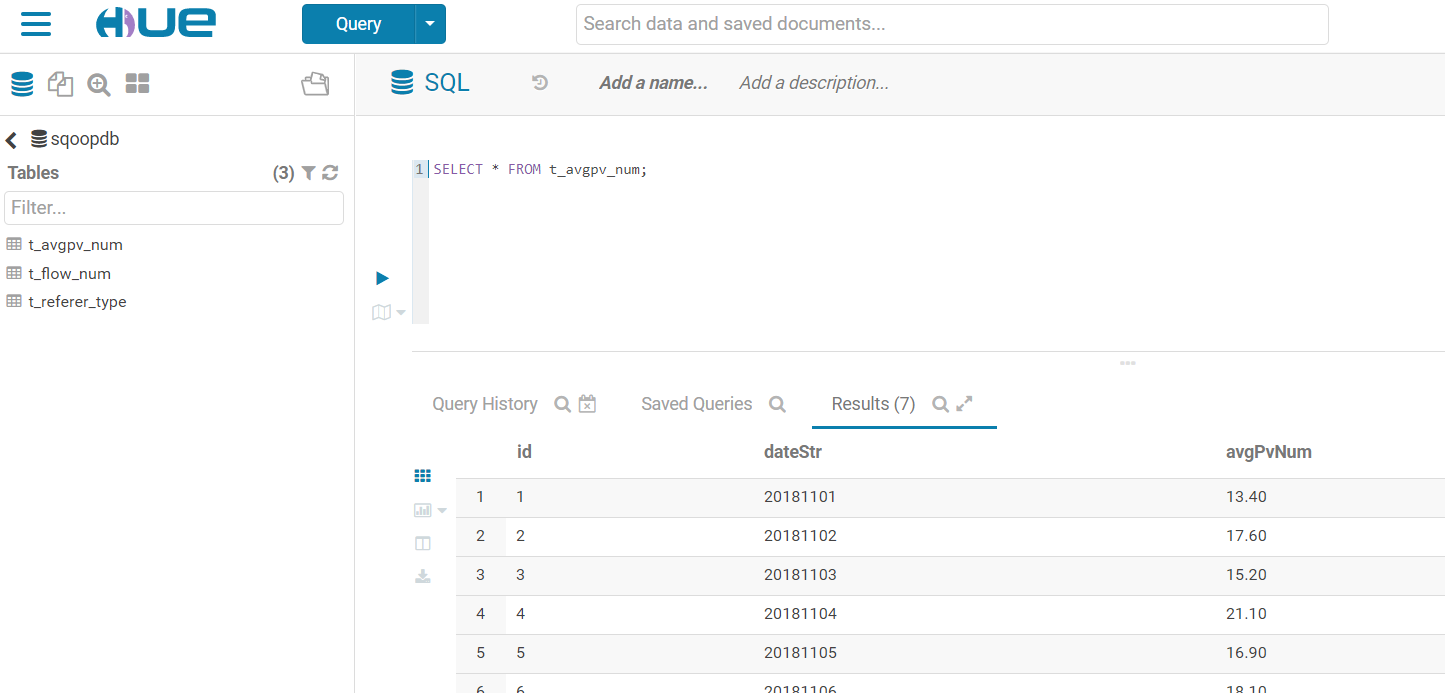
需要把mysql的注释给去掉。 大概位于1546行

|  |
| --- |
| [[[mysql]]]  nice\_name="My SQL DB"  engine=mysql  host=node-1  port=3306  user=root  password=hadoop |

### 重启hue

cd /export/servers/hue-3.9.0-cdh5.14.0/

build/env/bin/supervisor



## Hue集成Oozie

### 修改hue配置文件hue.ini

|  |
| --- |
| [liboozie]  # The URL where the Oozie service runs on. This is required in order for  # users to submit jobs. Empty value disables the config check.  oozie\_url=http://node-1:11000/oozie  # Requires FQDN in oozie\_url if enabled  ## security\_enabled=false  # Location on HDFS where the workflows/coordinator are deployed when submitted.  remote\_deployement\_dir=/user/root/oozie\_works |

|  |
| --- |
| [oozie]  # Location on local FS where the examples are stored.  # local\_data\_dir=/export/servers/oozie-4.1.0-cdh5.14.0/examples/apps  # Location on local FS where the data for the examples is stored.  # sample\_data\_dir=/export/servers/oozie-4.1.0-cdh5.14.0/examples/input-data  # Location on HDFS where the oozie examples and workflows are stored.  # Parameters are $TIME and $USER, e.g. /user/$USER/hue/workspaces/workflow-$TIME  # remote\_data\_dir=/user/root/oozie\_works/examples/apps  # Maximum of Oozie workflows or coodinators to retrieve in one API call.  oozie\_jobs\_count=100  # Use Cron format for defining the frequency of a Coordinator instead of the old frequency number/unit.  enable\_cron\_scheduling=true  # Flag to enable the saved Editor queries to be dragged and dropped into a workflow.  enable\_document\_action=true  # Flag to enable Oozie backend filtering instead of doing it at the page level in Javascript. Requires Oozie 4.3+.  enable\_oozie\_backend\_filtering=true  # Flag to enable the Impala action.  enable\_impala\_action=true |

|  |
| --- |
| [filebrowser]  # Location on local filesystem where the uploaded archives are temporary stored.  archive\_upload\_tempdir=/tmp  # Show Download Button for HDFS file browser.  show\_download\_button=true  # Show Upload Button for HDFS file browser.  show\_upload\_button=true  # Flag to enable the extraction of a uploaded archive in HDFS.  enable\_extract\_uploaded\_archive=true |

### 启动hue、oozie

启动hue进程

cd /export/servers/hue-3.9.0-cdh5.14.0

build/env/bin/supervisor

启动oozie进程

cd /export/servers/oozie-4.1.0-cdh5.14.0

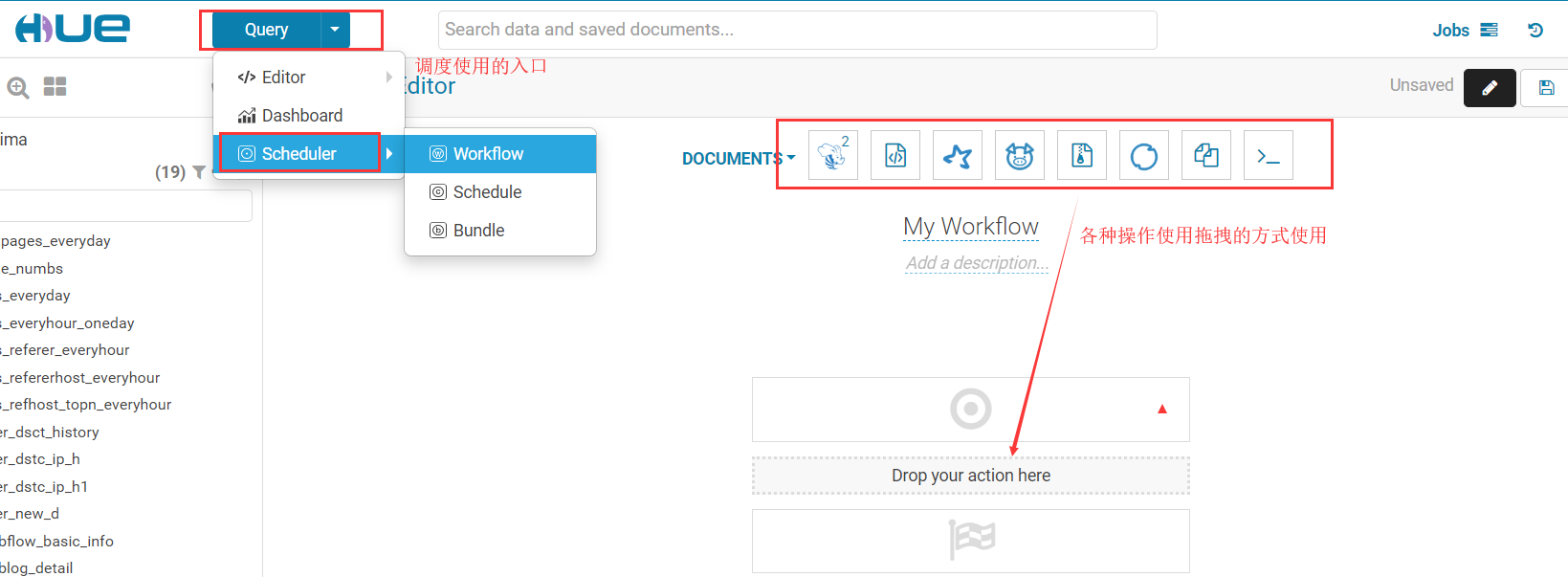
bin/oozied.sh start

页面访问hue

<http://node-1:8888/>

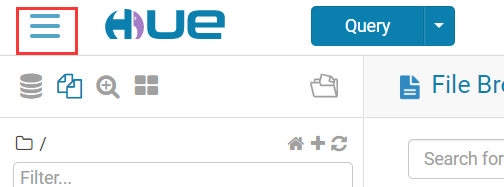
### 使用hue配置oozie调度

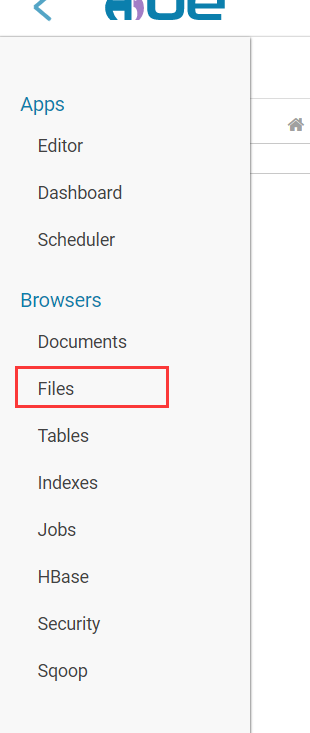
hue提供了页面鼠标拖拽的方式配置oozie调度



### 利用hue调度shell脚本

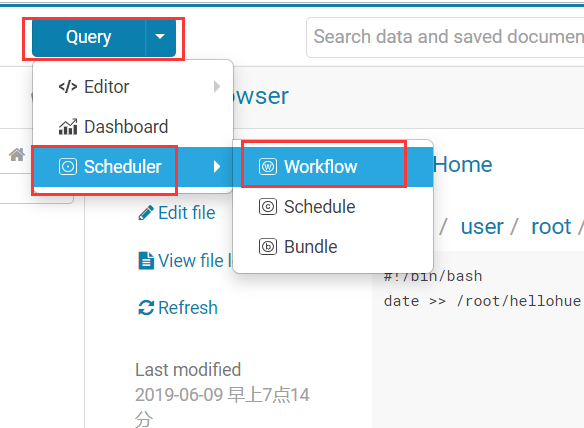
在HDFS上创建一个shell脚本程序文件。

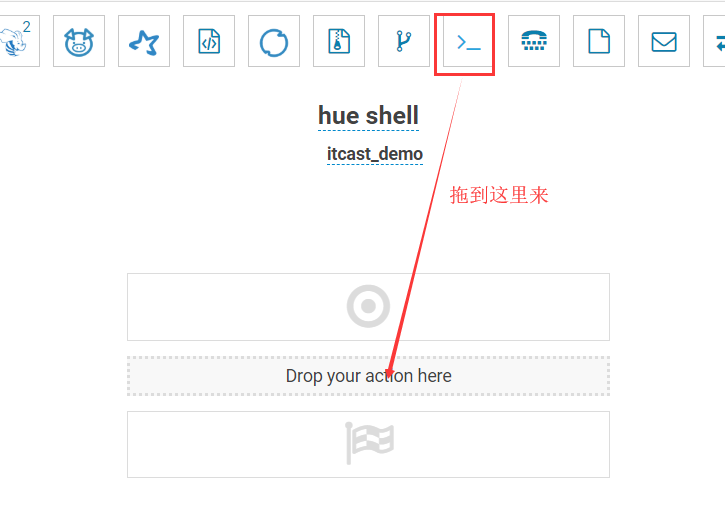


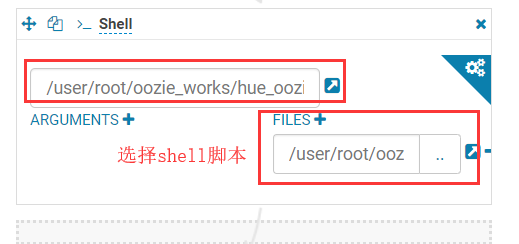




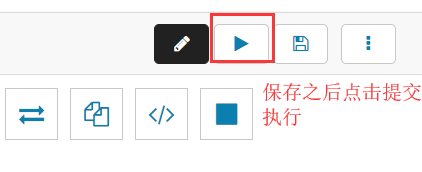
打开工作流调度页面。

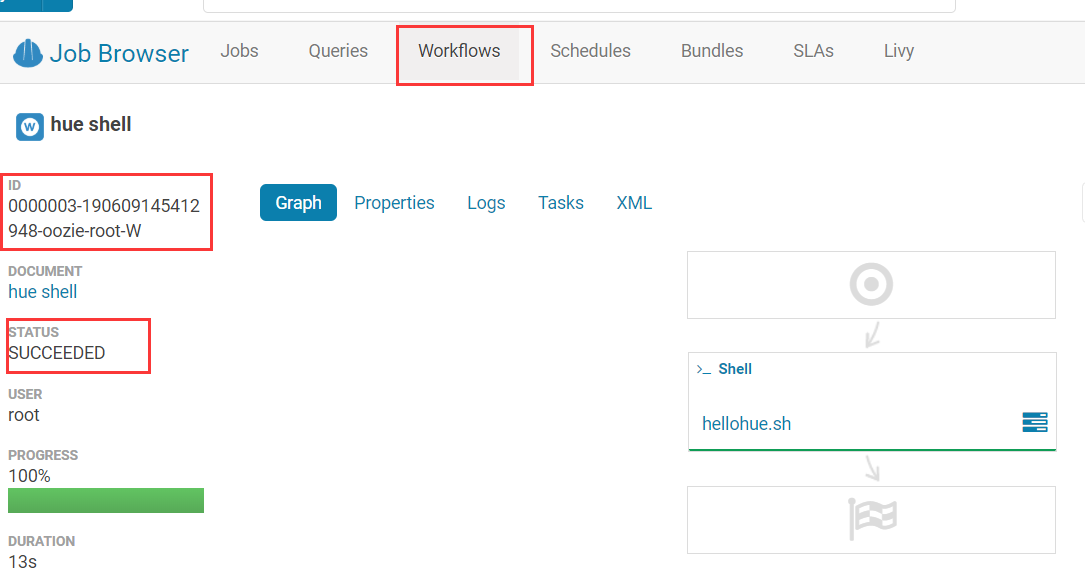










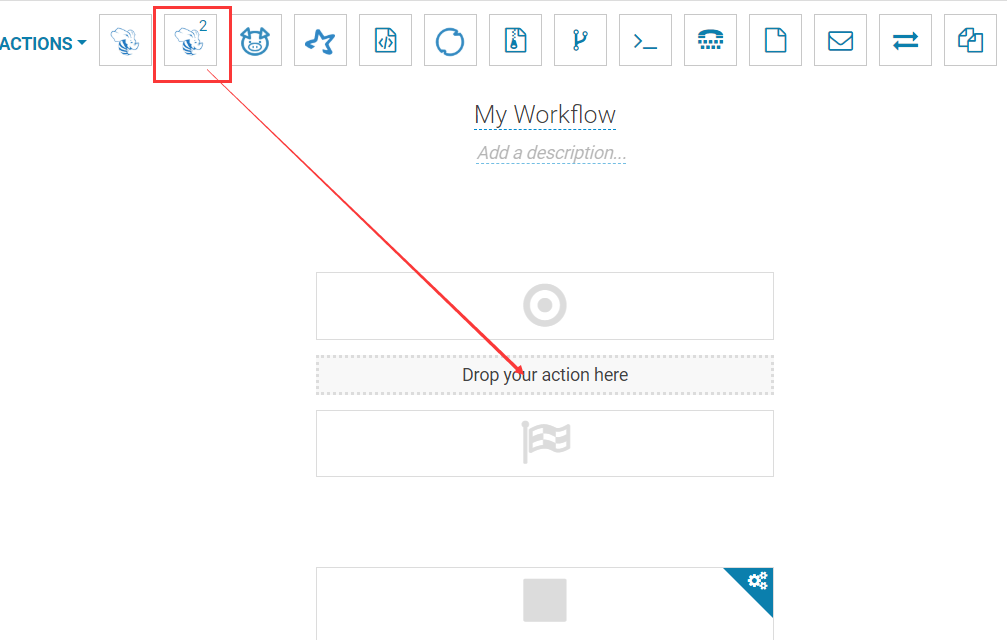


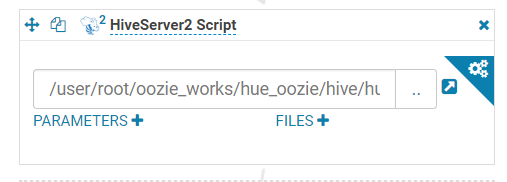
### 利用hue调度hive脚本

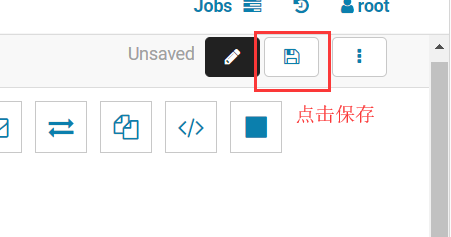
在HDFS上创建一个hive sql脚本程序文件。

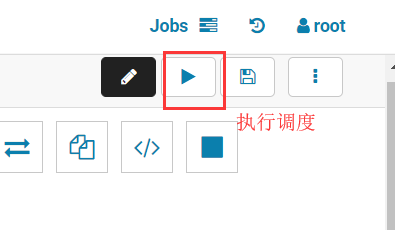


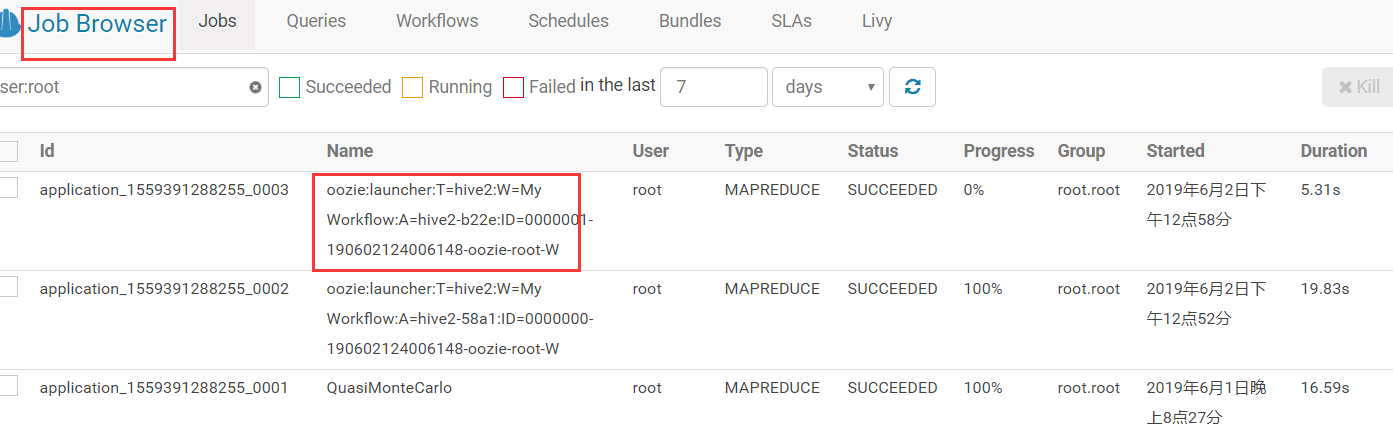
打开workflow页面，拖拽hive2图标到指定位置。





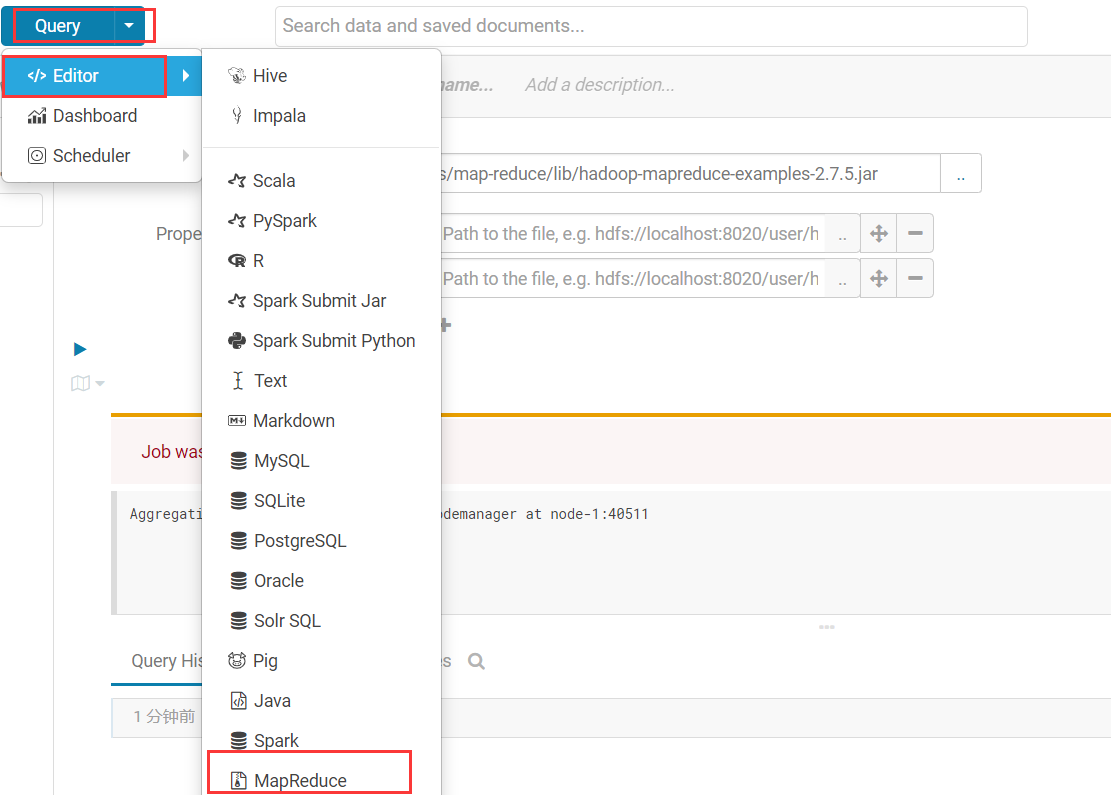


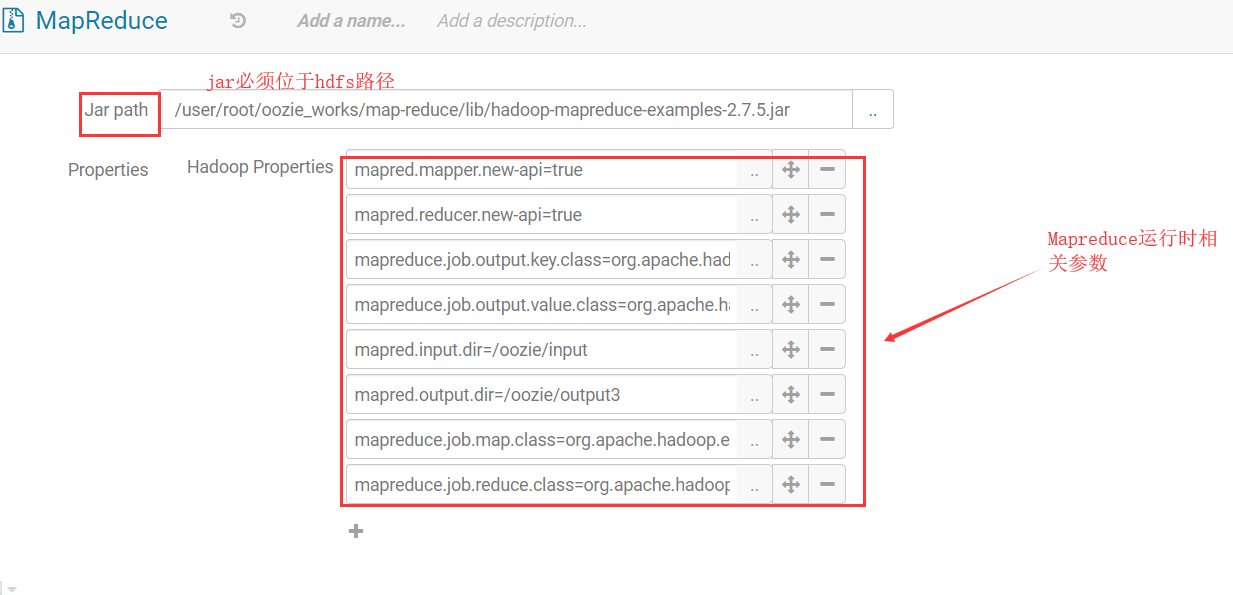


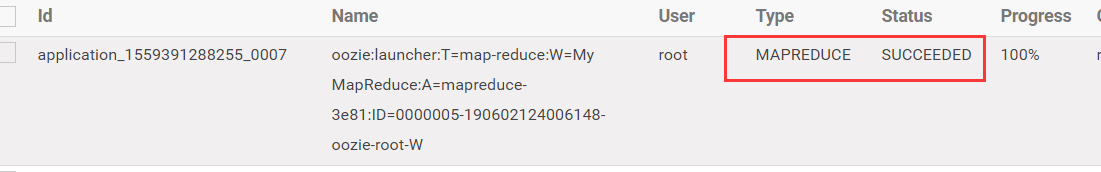


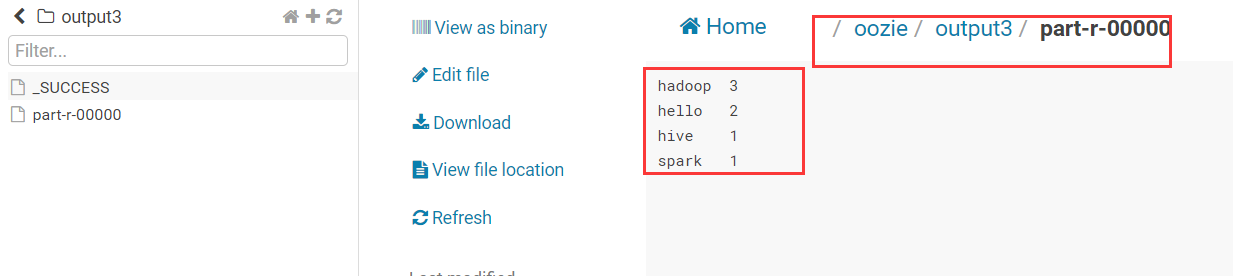
### 利用hue调度MapReduce程序

利用hue提交MapReduce程序



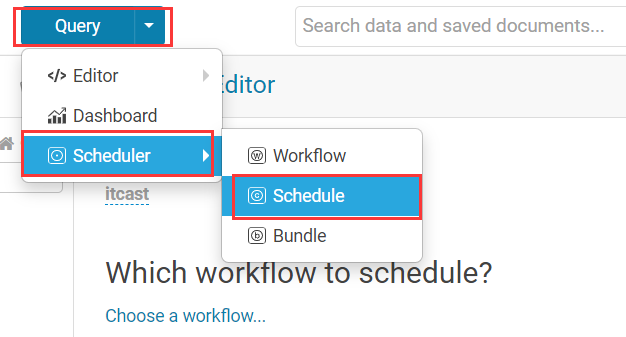


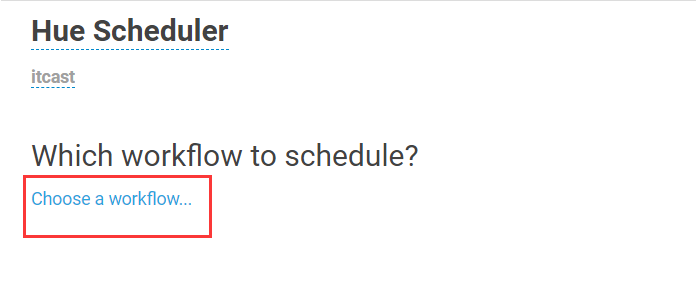


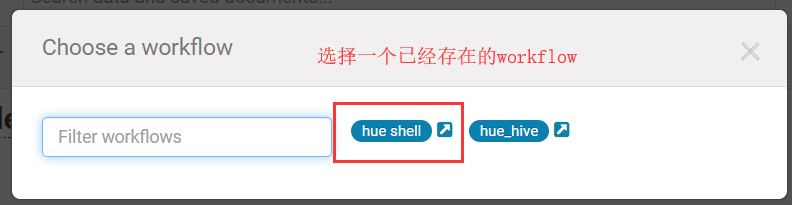


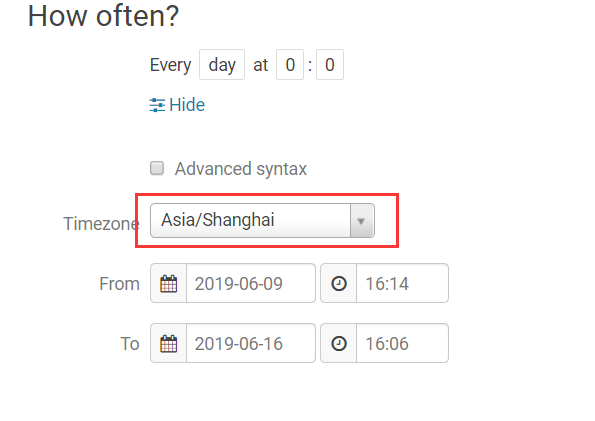
### 利用Hue配置定时调度任务

在hue中，也可以针对workflow配置定时调度任务，具体操作如下：

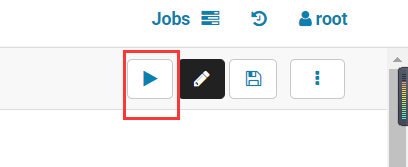


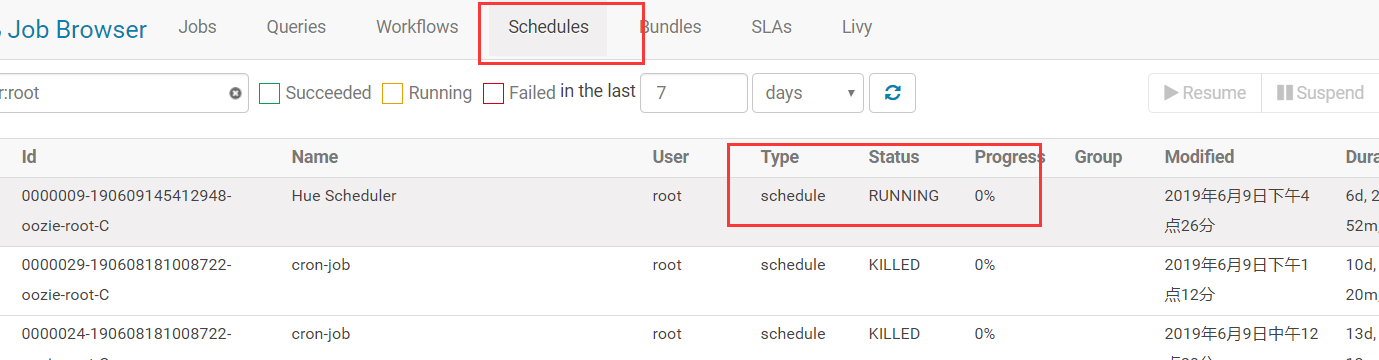




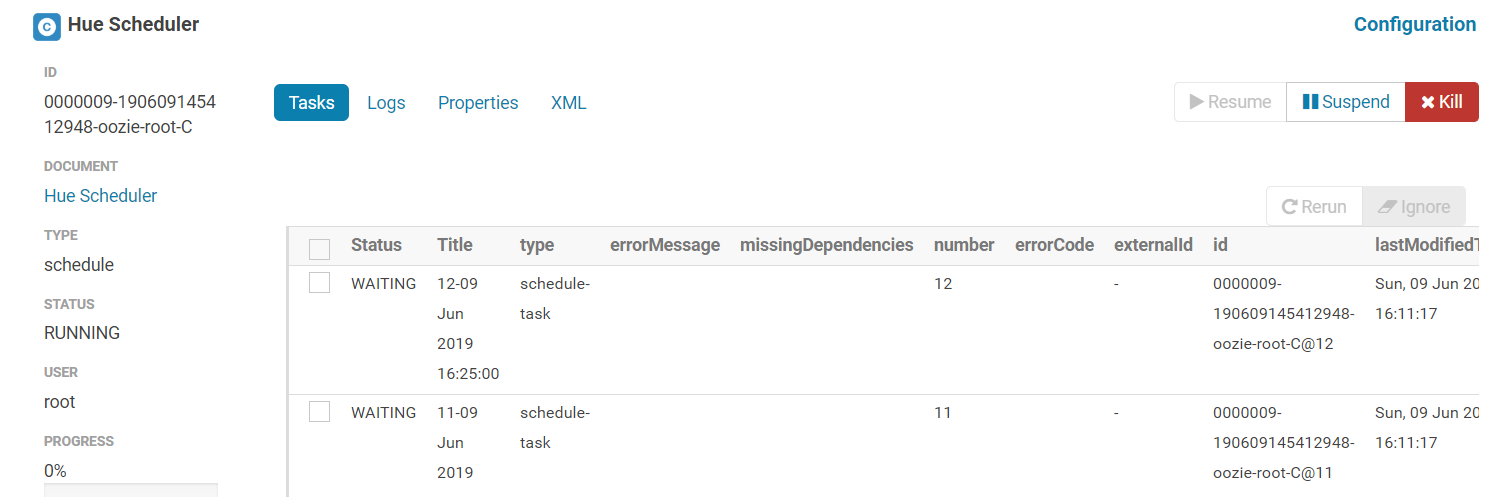


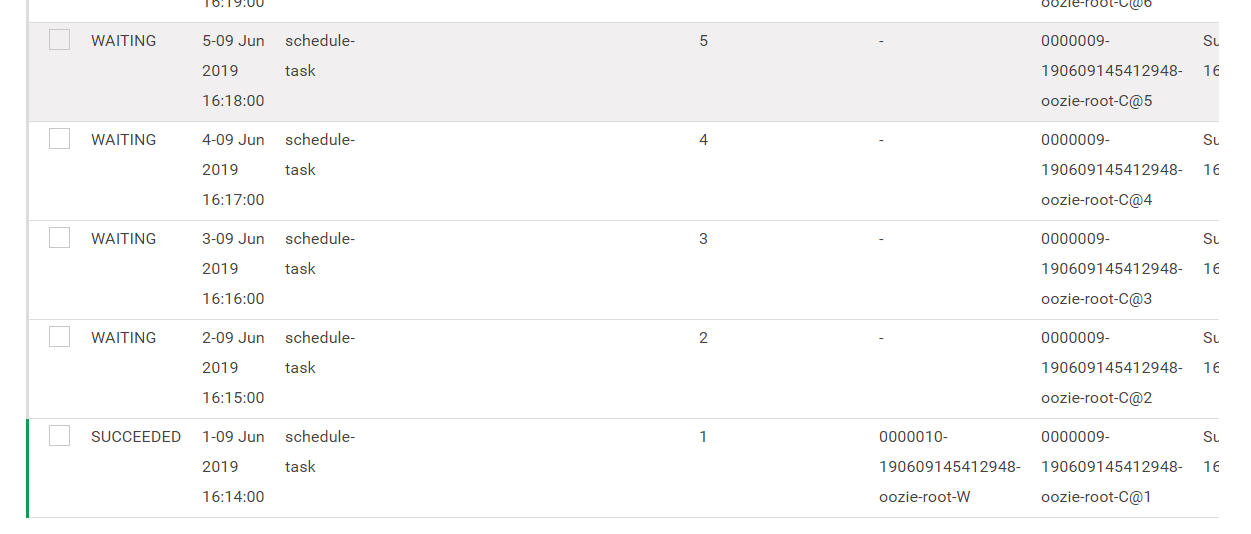
一定要注意时区的问题，否则调度就出错了。保存之后就可以提交定时任务。





点击进去，可以看到定时任务的详细信息。





## Hue集成Hbase

### 修改hbase配置

在hbase-site.xml配置文件中的添加如下内容，开启hbase thrift服务。

修改完成之后scp给其他机器上hbase安装包。

|  |
| --- |
| <property>    <name>hbase.thrift.support.proxyuser</name>    <value>true</value>  </property>  <property>    <name>hbase.regionserver.thrift.http</name>    <value>true</value>  </property> |

### 修改hadoop配置

在core-site.xml中确保 HBase被授权代理，添加下面内容。

把修改之后的配置文件scp给其他机器和hbase安装包conf目录下。

|  |
| --- |
| <property>  <name>hadoop.proxyuser.hbase.hosts</name>  <value>\*</value>  </property>  <property>  <name>hadoop.proxyuser.hbase.groups</name>  <value>\*</value>  </property> |

### 修改Hue配置

|  |
| --- |
| [hbase]  # Comma-separated list of HBase Thrift servers for clusters in the format of '(name|host:port)'.  # Use full hostname with security.  # If using Kerberos we assume GSSAPI SASL, not PLAIN.  hbase\_clusters=(Cluster|node-1:9090)  # HBase configuration directory, where hbase-site.xml is located.  hbase\_conf\_dir=/export/servers/hbase-1.2.1/conf  # Hard limit of rows or columns per row fetched before truncating.  ## truncate\_limit = 500  # 'buffered' is the default of the HBase Thrift Server and supports security.  # 'framed' can be used to chunk up responses,  # which is useful when used in conjunction with the nonblocking server in Thrift.  thrift\_transport=buffered |

### 启动hbase(包括thrift服务)、hue

需要启动hdfs和hbase，然后再启动thrift。

start-dfs.sh

start-hbase.sh

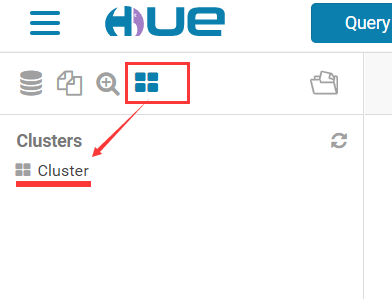
hbase-daemon.sh start thrift

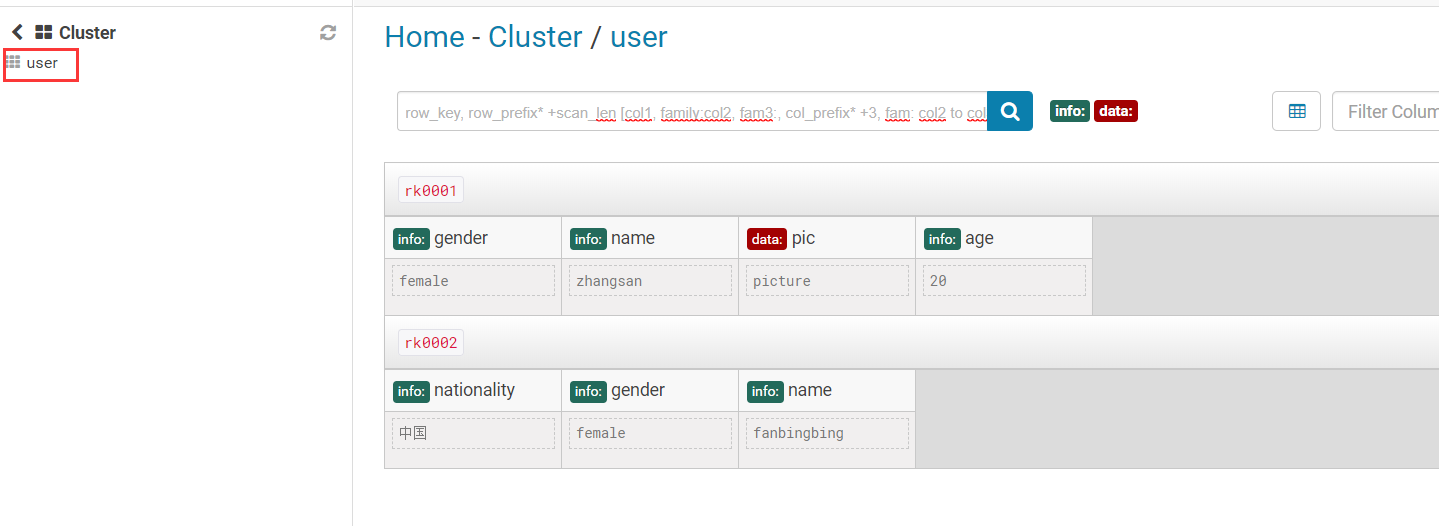
重新启动hue。

cd /export/servers/hue-3.9.0-cdh5.14.0/

build/env/bin/supervisor







## Hue集成Impala

### 修改Hue.ini

|  |
| --- |
| [impala]  server\_host=node-3  server\_port=21050  impala\_conf\_dir=/etc/impala/conf |

### 重启Hue

cd /export/servers/hue-3.9.0-cdh5.14.0/

build/env/bin/supervisor

