

# 尚硅谷大数据技术之 HUE

## 一、HUE 简介

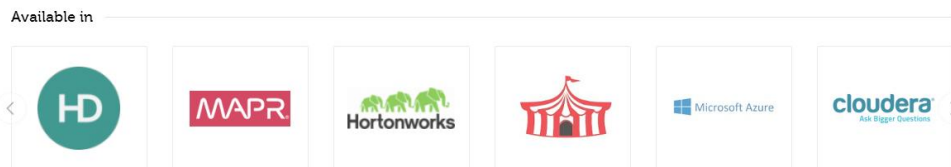
### 1.1、来源

HUE=Hadoop User Experience(Hadoop 用户体验), 直白来说就一个开源的 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。

### 1.2、官网及使用者

官网网站: <http://gethue.com/>

使用的公司:



## 二、HUE 安装（非 root 用户）

### 2.1、帮助文档

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

### 2.2、HUE 安装



HUE安装文档.do

c

## 三、HUE 与其他框架的集成

### 3.1、HUE 与 HDFS

【更多 Java、HTML5、Android、python、大数据 资料下载, 可访问尚硅谷(中国)官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

### 3.1.1、梳理集群环境

hadoop01	hadoop02	hadoop03	hadoop04
oozie			
zk	zk	zk	
hbase	hbase	hbase	
Hadoop	Hadoop	Hadoop	

### 3.1.2、配置 HDFS

修改: `hdfs-site.xml`

属性: `dfs.webhdfs.enabled`

属性值: `true`

解释: Enable WebHDFS (REST API) in Namenodes and Datanodes.

修改: `core-site.xml`

添加

```
<property>
```

```
  <name>hadoop.proxyuser.hue.hosts</name>
```

```
  <value>*</value>
```

```
</property>
```

```
<property>
```

```
  <name>hadoop.proxyuser.hue.groups</name>
```

```
  <value>*</value>
```

```
</property>
```

#设置代理用户

```
<property>
```

```
  <name>hadoop.proxyuser.hadoop.hosts</name>
```

```
  <value>*</value>
```

```
</property>
```

【更多 Java、HTML5、Android、python、大数据 资料下载，可访问尚硅谷（中国）官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

```
<property>

  <name>hadoop.proxyuser.hadoop.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。（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>
```

区别：WebHDFS 是 HDFS 内置的组件，已经运行于 NameNode 和 DataNode 中。对 HDFS 文件的读写，将会重定向到文件所在的 DataNode，并且会完全利用 HDFS 的带宽。HttpFS 是独立于 HDFS 的一个服务。对 HDFS 文件的读写，将会通过它进行中转，它能限制带宽占用。

#### 修改：httpfs-site.xml

```
<property>

  <name>httpfs.proxyuser.hue.hosts</name>

  <value>*</value>

</property>

<property>

  <name>httpfs.proxyuser.hue.groups</name>

  <value>*</value>

</property>
```

解释：以上两个属性主要用于 HUE 服务与 Hadoop 服务不在同一台节点上所必须的配置。

提示：

【更多 Java、HTML5、Android、python、大数据 资料下载，可访问尚硅谷（中国）官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

- \* 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

### 3.1.3、scp 同步配置

```
//打包
$ tar cvf conf.tar.gz  hadoop/

//发送
$ scp -r conf.tar.gz  hadoop@192.168.56.87:/home/hadoop/apps/hadoop26/etc/
$ scp -r conf.tar.gz  hadoop@192.168.56.87:/home/hadoop/apps/hadoop26/etc/
```

### 3.1.4、启动 httpfs 服务

```
$ /home/hadoop/apps/hadoop26/sbin/httpfs.sh start
```

### 3.1.5、配置 hue.ini 文件

找到[hadoop]标签

```
[[hdfs_clusters]]

# HA support by using HttpFs

[[[default]]]

# Enter the filesystem uri

##HDFS 服务器地址

fs_defaultfs=hdfs://192.168.56.86:8020

# 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

##向 HDFS 发送命令的请求地址

webhdfs_url=http://192.168.56.86:14000/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 的一些配置

hadoop_conf_dir=/home/hadoop/apps/hadoop26/etc/hadoop

hadoop_hdfs_home=/home/hadoop/apps/hadoop26

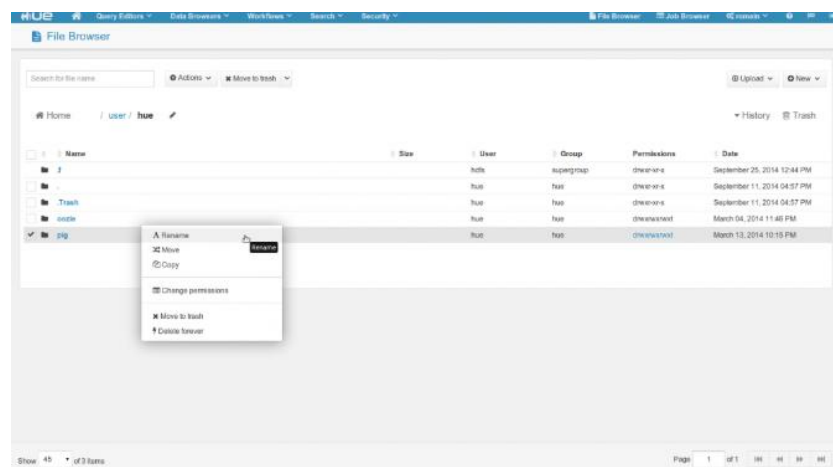
hadoop_bin=/home/hadoop/apps/hadoop26/bin
```

### 3.1.6、测试

开启 HUE 服务：

```
$ build/env/bin/supervisor
```

打开 HUE 的页面，进行 HDFS 管理。



**提示 1**(不修改也可以):

如果提示错误根目录应该归属于 hdfs，如下图：

【更多 Java、HTML5、Android、python、大数据 资料下载，可访问尚硅谷（中国）官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

```
hadoop.hdfs_clusters.default.webhdfs_url 当前值: http://192.168.56.86:14000/webhdfs/v1
文件系统根目录"/"应归属于"hdfs"
```

请修改 python 变量，位置如下

```
cd desktop/libs/hadoop/src/hadoop/fs/
vi webhdfs.py
修改其中的变量值为:
DEFAULT_HDFS_SUPERUSER = 'hadoop' #设置的 hue 超级用户
```

然后重启 HUE 服务即可。

#### 提示 2:

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：

```
$ netstat -tunlp | grep 8888
```

## 3.2、HUE 与 YARN

### 3.2.1、配置 hue.ini

找到[[yarn\_clusters]]标签，涉及修改配置如下：

```
[[yarn_clusters]]
[[[default]]]
#yarn 服务的配置
resourcemanager_host=192.168.56.86
resourcemanager_port=8032
#是否将作业提交到此群集，并监控作业执行情况
submit_to=True
#logical_name=cluster-yarn1(如果开高可用的话)
#配置 yarn 资源管理的访问入口
resourcemanager_api_url=http://192.168.56.86:8088
proxy_api_url=http://192.168.56.86:8088
#历史服务器管理的入口，查看作业的历史运行情况
history_server_api_url=http://192.168.56.87:19888
```

Hadoop 启动 jobhistoryserver 来实现 web 查看作业的历史运行情况: `./mr-jobhistory-daemon.sh start historyserver`

### 3.2.2、重启 HUE 测试

```
$ build/env/bin/supervisor  
  
//测试  
  
./bin/yarn jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.1.jar wordcount /input/ /output
```

## 3.3、HUE 与 Hive

### 3.3.1、修改 Hive 配置文件 `apache-hive-2.3.3-bin/conf/hive-site.xml`

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

属性: `hive.server2.thrift.port`  
属性值: 10000  
描述: TCP 的监听端口

属性: `hive.server2.thrift.bind.host`  
属性值: 192.168.56.88  
描述: TCP 绑定的主机

属性: `hive.server2.long.polling.timeout`  
属性值: 5000  
描述: HiveServer2 在响应使用长轮询的异步调用之前等待的时间（毫秒）

属性: `hive.metastore.uris`  
属性值: `thrift://192.168.56.88:9083`  
描述: 指向的是运行 metastore 服务的主机

### 3.3.2、启动 Hive 相关服务

```
$ bin/hive --service metastore &
```

【更多 Java、HTML5、Android、python、大数据 资料下载，可访问尚硅谷（中国）官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

```
$ bin/hive --service hiveserver2 &
```

**提示 1:** 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。

**提示 2:** 如果如法正确启动，请尝试 kill -9 RunJar ，在重新启动

### 3.3.3、配置 hue.ini

找到[beeswax]属性标签，涉及修改如下：

```
[beeswax]

hive_server_host=192.168.56.88

hive_server_port=10000

hive_conf_dir=/home/hadoop/apps/apache-hive-2.3.3-bin/conf
```

### 3.3.4、重启 HUE 测试

```
$ build/env/bin/supervisor
```

## 3.4、HUE 与 Mysql

### 3.4.1、配置 hue.ini

找到[[[mysql]]]标签，涉及修改如下：

```
[[[mysql]]]

nice_name=db_mysql

engine=mysql

host=192.168.56.88

port=3306

user=root

password=123456
```

### 3.4.2、重启 hue.ini 测试

启动后即可测试是否成功连接 Mysql 服务，并且测试是否可以查看数据

```
$ build/env/bin/supervisor
```

【更多 Java、HTML5、Android、python、大数据 资料下载，可访问尚硅谷（中国）官网 [www.atguigu.com](http://www.atguigu.com) 下载区】



### 3.5.3、重启 HUE 测试

```
$ build/env/bin/supervisor
```

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：

```
$ bin/hdfs dfs -mkdir -p /user/oozie/share/lib
```

## 3.5、HUE 与 Zookeeper

### 3.7.1、配置 hue.ini

找到[zookeeper]标签，涉及修改如下：

```
[zookeeper]

[[clusters]]

    [[[default]]]

        host_ports=192.168.56.86:2181,192.168.56.87:2181,192.168.56.88:2181
```

### 3.7.2、启动 zk 测试

```
启动：
zkServer.sh start
状态：
zkServer.sh status
```

### 3.7.3、重启 HUE 测试

```
$ build/env/bin/supervisor
```

## 3.6、HUE 与 HBase(先启动 zk)

### 3.6.1、修改 hue.ini 配置

找到[hbase]标签，涉及修改内容如下：

```
[hbase]

hbase_clusters=(Cluster|192.168.56.87:9090)

hbase_conf_dir=/home/hadoop/apps/hbase/conf
```

【更多 Java、HTML5、Android、python、大数据 资料下载，可访问尚硅谷（中国）官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

### 3.6.2、启动 HBase 服务和 Hbase 的 thrift 服务

```
启动:  
$ bin/start-hbase.sh  
$ bin/hbase-daemon.sh start thrift
```

### 3.6.3、重启 HUE 进行测试

```
$ build/env/bin/supervisor
```

## 3.7、HUE 与 Oozie

### 3.5.1、配置 hue.ini

找到[liboozie]标签以及[oozie]标签涉及修改如下:

[liboozie]:

```
[liboozie]  
  
#运行 Oozie 服务的 URL。  
  
oozie_url=http://192.168.56.87:11000/oozie  
  
#HDFS 上的位置，提交时部署工作流。  
  
remote_deployment_dir=/user/hadoop/oozie-apps
```

[oozie]:

```
[oozie]  
  
#本地存储 oozie 示例的位置。  
  
local_data_dir=/home/hadoop/apps/oozie-4.0.0-cdh5.3.6/examples  
  
#本地存储 oozie 示例数据的位置。  
  
sample_data_dir=/home/hadoop/apps/oozie-4.0.0-cdh5.3.6/oozie-apps  
  
#HDFS 上存储 oozie 示例和工作流的位置。  
  
remote_data_dir=/user/hadoop/oozie-apps  
  
#启用 Cron 来定义协调器的频率替换旧的频率数/单位。  
  
enable_cron_scheduling=true
```

### 3.5.2、启动 Oozie 相关服务

```
$ bin/oozied.sh start
```

【更多 Java、HTML5、Android、python、大数据 资料下载，可访问尚硅谷（中国）官网 [www.atguigu.com](http://www.atguigu.com) 下载区】

## 四、总结

在此我们总结一下集成 HUE 时，我们开启的后台服务项

### 4.1、Hadoop (hdfs 和 yarn)

启动：

```
$ ./bin/start-hdfs.sh  
$ ./bin/start-hdfs.sh  
或  
$ ./bin/start-all.sh
```

注意：

```
$ ~/modules/cdh/hadoop-2.5.0-cdh5.3.6/sbin/httpfs.sh start
```

### 4.2、Hive

启动：

```
$ ~/modules/cdh/hive-0.13.1-cdh5.3.6/bin/hive --service metastore  
$ ~/modules/cdh/hive-0.13.1-cdh5.3.6/bin/hive --service hiveserver2  
或  
$ ./bin/hive
```

### 4.3、zk 启动

启动：

```
zkServer.sh start
```

状态：

```
zkServer.sh status
```

### 4.4、HBase

启动：

```
bin/start-hbase.sh
```

注意：在 192.168.56.87 thrift

```
$ bin/hbase-daemon.sh start thrift &
```

## 4.5、Oozie

启动：

```
$ bin/oozied.sh start &
```

注意：

先启动 **hadoop** 和 **zookeeper**