**Hue安装配置实践**

Hue是一个开源的Apache Hadoop UI系统，最早是由Cloudera Desktop演化而来，由Cloudera贡献给开源社区，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，例如操作HDFS上的数据，运行MapReduce Job等等。很早以前就听说过Hue的便利与强大，一直没能亲自尝试使用，下面先通过官网给出的特性，通过翻译原文简单了解一下Hue所支持的功能特性集合：

默认基于轻量级sqlite数据库管理会话数据，用户认证和授权，可以自定义为MySQL、Postgresql，以及Oracle

基于文件浏览器（File Browser）访问HDFS

基于Hive编辑器来开发和运行Hive查询

支持基于Solr进行搜索的应用，并提供可视化的数据视图，以及仪表板（Dashboard）

支持基于Impala的应用进行交互式查询

支持Spark编辑器和仪表板（Dashboard）

支持Pig编辑器，并能够提交脚本任务

支持Oozie编辑器，可以通过仪表板提交和监控Workflow、Coordinator和Bundle

支持HBase浏览器，能够可视化数据、查询数据、修改HBase表

支持Metastore浏览器，可以访问Hive的元数据，以及HCatalog

支持Job浏览器，能够访问MapReduce Job（MR1/MR2-YARN）

支持Job设计器，能够创建MapReduce/Streaming/Java Job

支持Sqoop 2编辑器和仪表板（Dashboard）

支持ZooKeeper浏览器和编辑器

支持MySql、PostGresql、Sqlite和Oracle数据库查询编辑器

软件下载：

<http://archive.cloudera.com/cdh5/cdh/5/hue-3.9.0-cdh5.13.1.tar.gz>

安装和功能介绍参考：

<https://github.com/cloudera/hue>

下载软件：

# wget <http://archive.cloudera.com/cdh5/cdh/5/hue-3.9.0-cdh5.13.1.tar.gz>

解压：

# tar -zxf hue-3.9.0-cdh5.13.1.tar.gz

安装，在安装之前我们需要安装一下maven环境，应为我们的hue依赖maven环境，因此编译的时候依赖这个

# wget <http://mirrors.tuna.tsinghua.edu.cn/apache/maven/maven-3/3.5.2/binaries/apache-maven-3.5.2-bin.tar.gz>

解压：

# tar -zxf apache-maven-3.5.2-bin.tar.gz -C /usr/local/

配置maven的环境变量：

#maven env

MAVEN\_HOME=/usr/local/apache-maven-3.5.2

PATH=$MAVEN\_HOME/bin:$PATH

刷新配置文件，是配置生效：

测试：

[root@hadoop01 source]# mvn -v

Apache Maven 3.5.2 (138edd61fd100ec658bfa2d307c43b76940a5d7d; 2017-10-18T15:58:13+08:00)

Maven home: /usr/local/apache-maven-3.5.2

Java version: 1.8.0\_111, vendor: Oracle Corporation

Java home: /usr/local/java/jdk1.8.0\_111/jre

Default locale: en\_US, platform encoding: UTF-8

OS name: "linux", version: "2.6.32-431.el6.x86\_64", arch: "amd64", family: "unix"

开始编译安装hue了，安装依赖关系：

#yum install -y asciidoc cyrus-sasl-devel cyrus-sasl-gssapi gcc gcc-c++ krb5-devel libtidy libxml2-devel libxslt-devel make mysql mysql-devel openldap-devel python-devel sqlite-devel openssl-devell2-devel libxslt-devel make mysql mysql-devel openldap-devel python-devel sqlite-devel openssl-devel gmp-devel

切换目录

# cd hue-3.9.0-cdh5.13.1

安装

# make apps

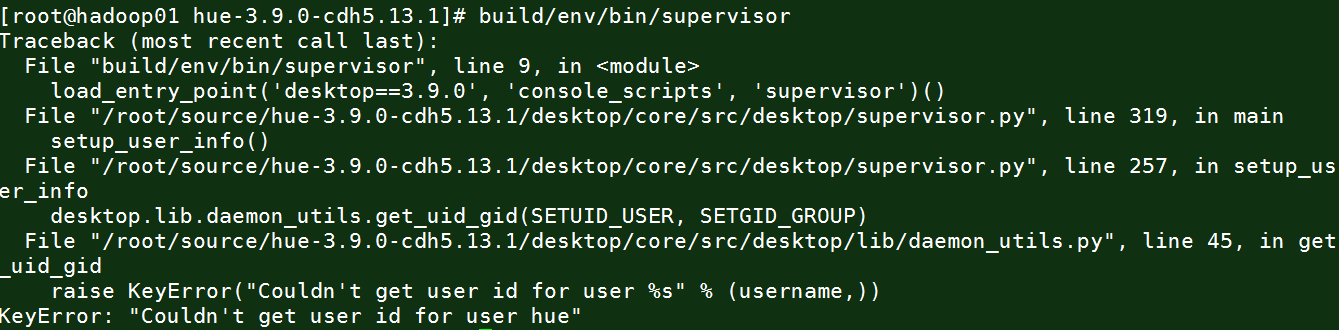
遇到报错：

gcc: Internal error: Killed (program cc1)

这个问题是内存不足导致的报错，升级一下内存再次编译就可以了

启动hue如下命令：

[root@hadoop01 hue-3.9.0-cdh5.13.1]# build/env/bin/supervisor



报错，是因为我们的系统中没有hue用户导致的，我们可以通过修改配置文件或是创建这个用户进行解决：

修改配置文件，将启动的用户修改为root:

[root@hadoop01 conf]# pwd

/root/source/hue-3.9.0-cdh5.13.1/desktop/conf

Vim hue.ini

## server\_user=hue

## server\_group=hue

修改为：

server\_user=root

server\_group=root

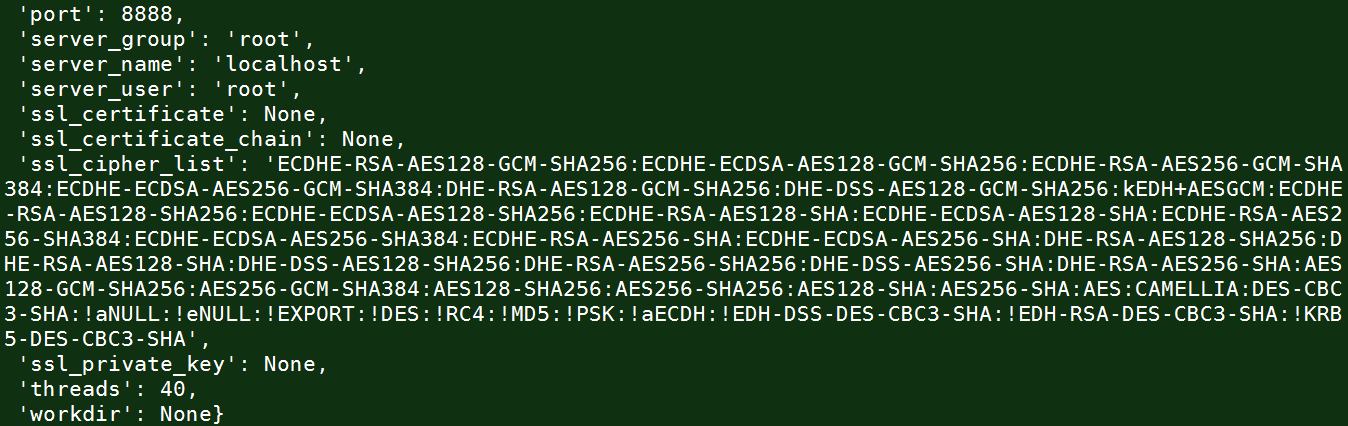
default\_user=root

最后我们在创建一个用户hue

Useradd hue

尝试再次启动：

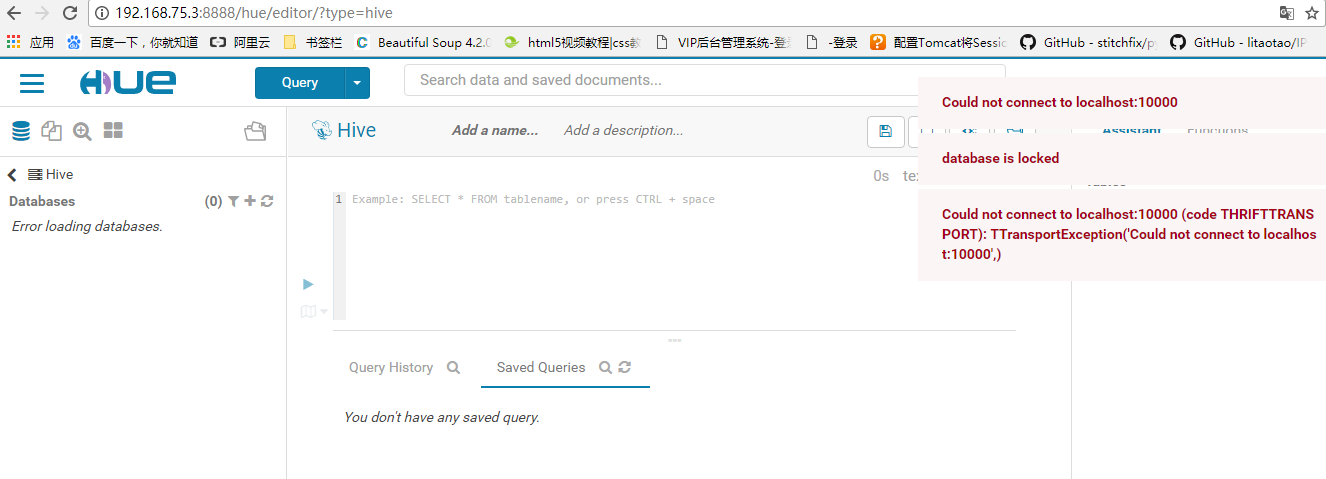
[root@hadoop01 hue-3.9.0-cdh5.13.1]# build/env/bin/supervisor



我们可以通过浏览器访问我们的hue的UI界面。访问地址为：

<http://192.168.75.3:8888>

访问出现登录界面，用户密码都是admin 进入后界面如下：



为了保证程序正常的启动和运行我们需要修改一下权限：

[root@hadoop01 hue-3.9.0-cdh5.13.1]# chmod 777 desktop/

修改db文件

[root@hadoop01 hue-3.9.0-cdh5.13.1]# chmod 766 desktop/desktop.db

**二、集成hadoop**

参考官方文档地址：

<http://hadoop.apache.org/docs/r2.7.3/hadoop-project-dist/hadoop-common/>

hadoop单节点配置：

1. 解压我们的hadoop

[root@hadoop01 ~]# cd source/

[root@hadoop01 source]# tar -zxf hadoop-2.7.3.tar.gz -C /opt/

切换到hadoop的配置目录中：

[root@hadoop01 hadoop]# pwd

/opt/hadoop-2.7.3/etc/hadoop

参考官方文档进行配置文件的修改：

<http://hadoop.apache.org/docs/r2.7.3/hadoop-project-dist/hadoop-common/SingleCluster.html>

修改etc/hadoop/hadoop-env.sh

将

#export JAVA\_HOME=${JAVA\_HOME}

修改为

export JAVA\_HOME=/usr/local/java/jdk1.8.0\_111

修改etc/hadoop/core-site.xml:

<property>

<name>fs.defaultFS</name>

<value>hdfs://localhost:8020</value>

</property>

<property>

<name>hadoop.tmp.dir</name>

<value>/opt/hadoop-2.7.3/data</value>

</property>

修改etc/hadoop/hdfs-site.xml:

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

配置修改完成后，格式化一下：

[root@hadoop01 ~]# cd /opt/hadoop-2.7.3/

[root@hadoop01 hadoop-2.7.3]# bin/hdfs namenode -format

继续配置：

修改etc/hadoop/mapred-site.xml:

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

修改etc/hadoop/yarn-site.xml:

<property>

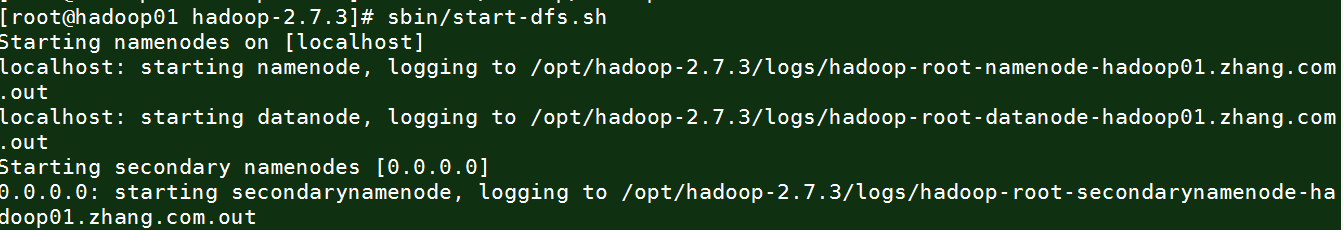
<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

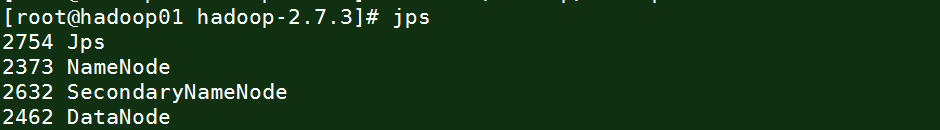
</property>

分别启动hdfs和yarn

[root@hadoop01 hadoop-2.7.3]# sbin/start-dfs.sh

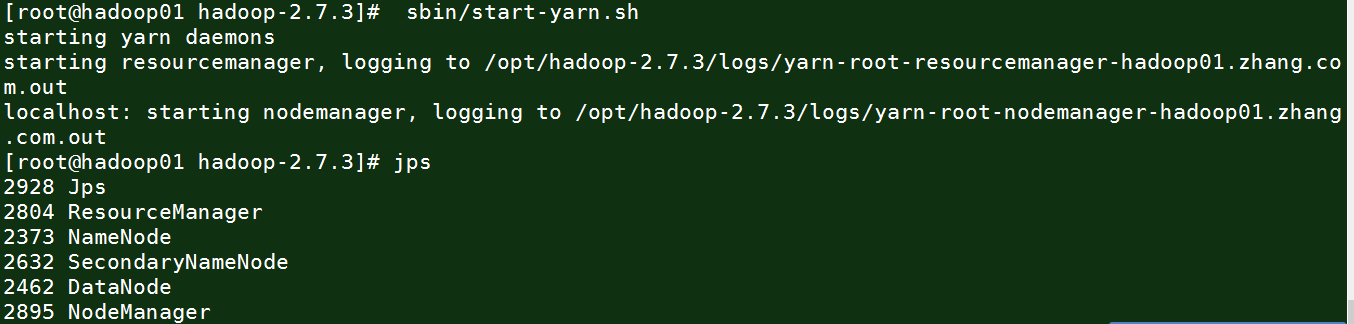


使用jps验证一下：



启动yarn:

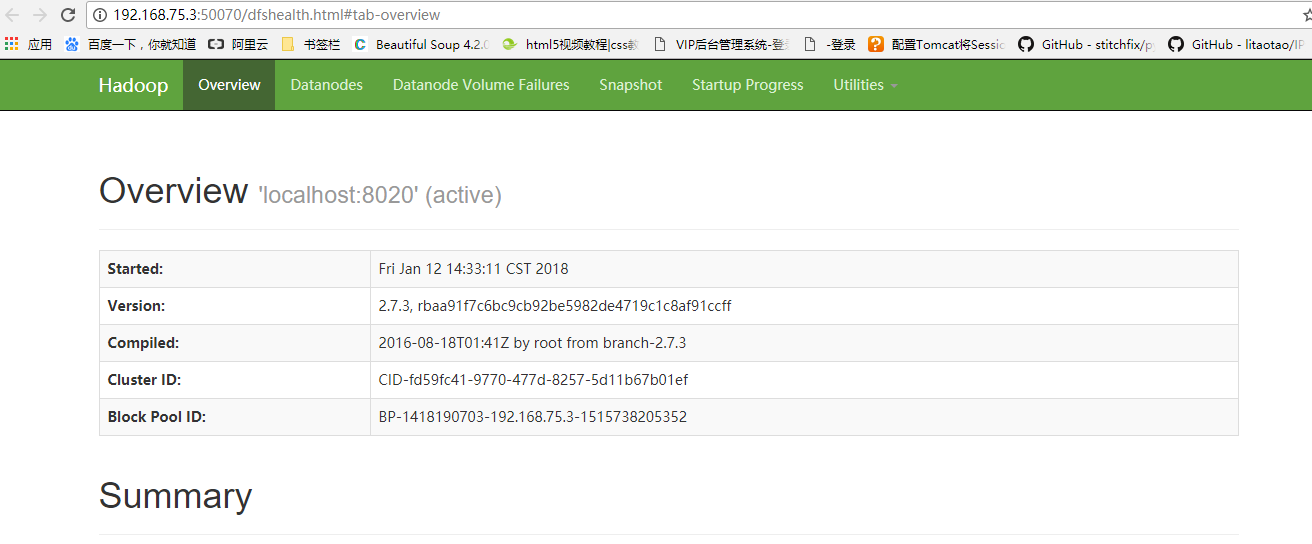
# sbin/start-yarn.sh

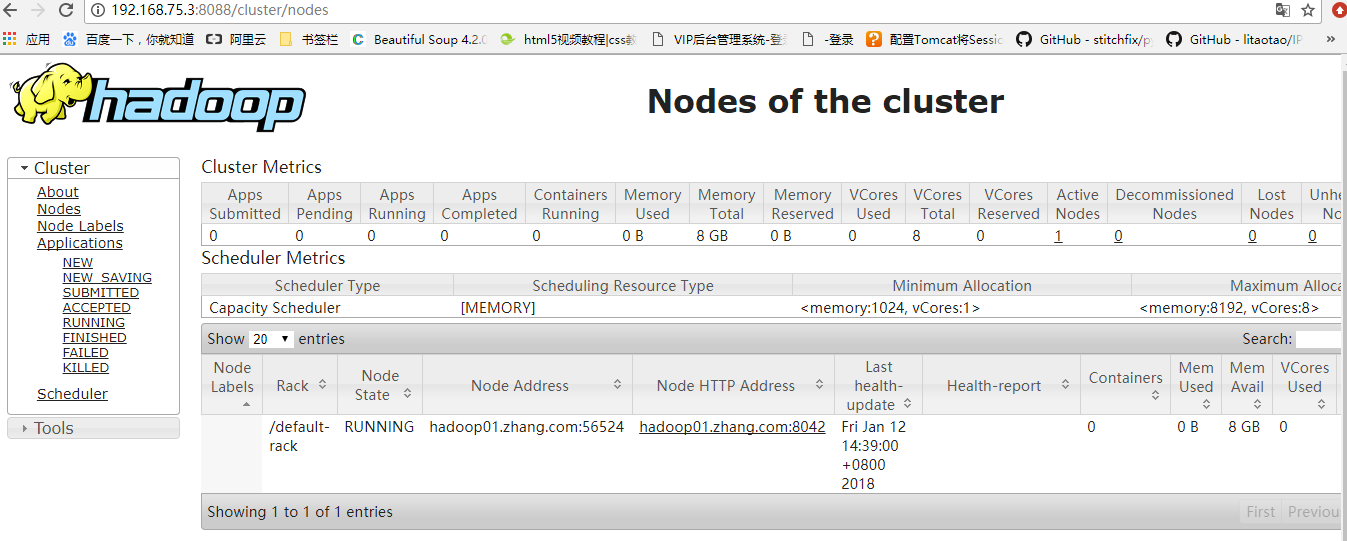


我们依次访问hdfs和yarn的web界面看一下：

访问：<http://192.168.75.3:50070/>

http://192.168.75.3:8088





核实没有问题后。创建一个测试文件上传：

[root@hadoop01 hadoop-2.7.3]# cat words

hello a

hello b

hello c

hello d

上传文件：

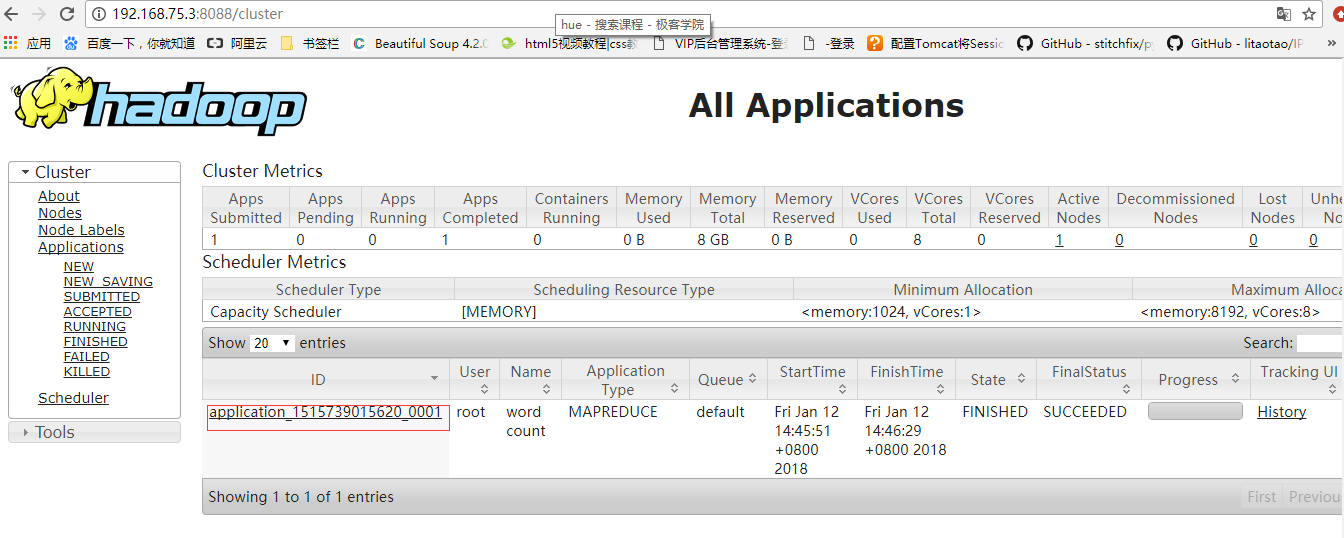
[root@hadoop01 hadoop-2.7.3]# bin/hdfs dfs -put words /words

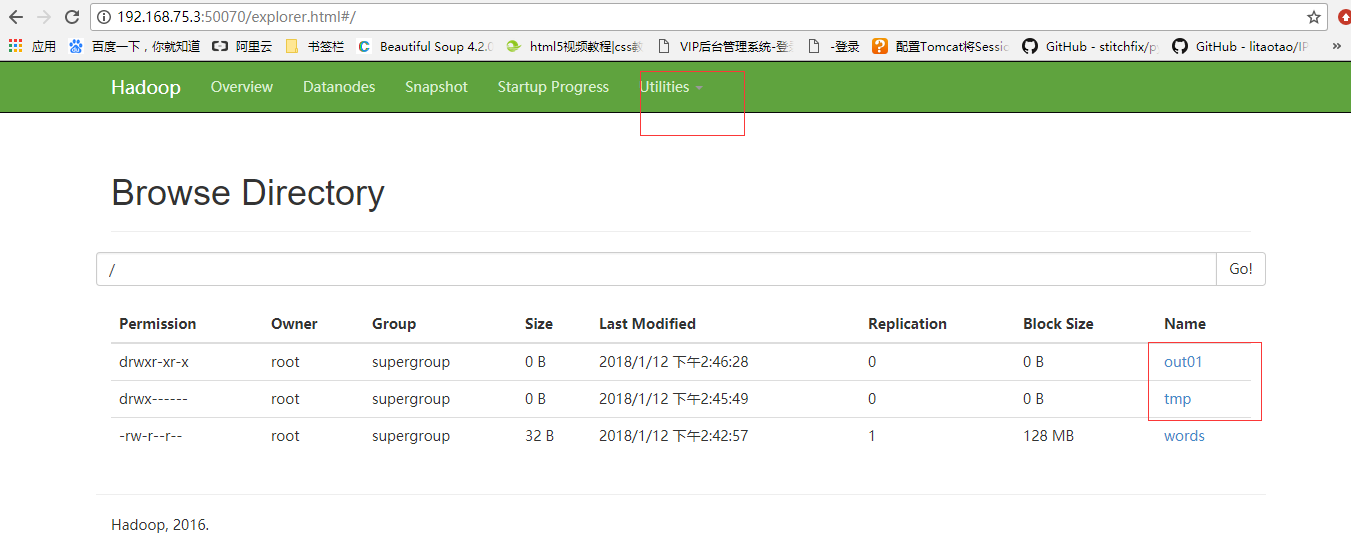
运行统计词频的命令：

[root@hadoop01 hadoop-2.7.3]#

bin/yarn jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar wordcount /words /out01

运行完成后我们可以在web界面中查看一下，会在hdfs中多出一个文件，同时会在yarn中多出一个完成的任务

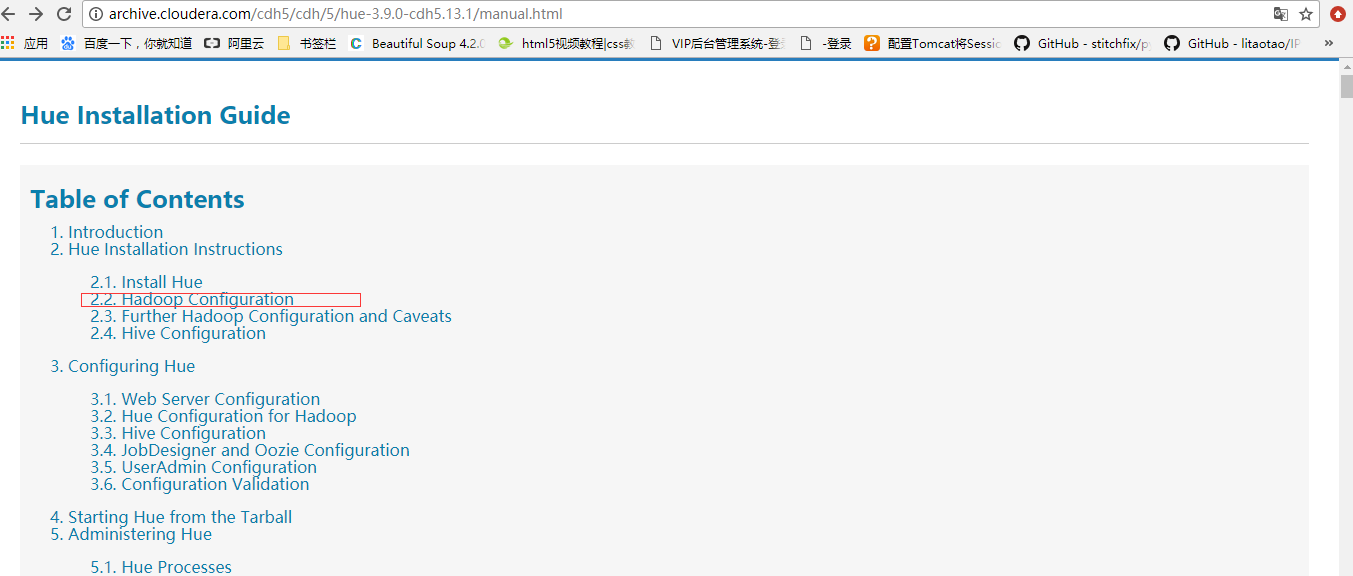


  
整合hadoop和hue

参看hue的官方文档：

http://archive.cloudera.com/cdh5/cdh/5/hue-3.9.0-cdh5.13.1/manual.html

开始配置：



[root@hadoop01 hadoop-2.7.3]# vim etc/hadoop/hdfs-site.xml

<!-- 开启webhdfs的restful的api-->

<property>

<name>dfs.webhdfs.enabled</name>

<value>true</value>

</property>

[root@hadoop01 hadoop-2.7.3]# vim etc/hadoop/core-site.xml

<!—制定代理的用户-->

<property>

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

<value>\*</value>

</property>

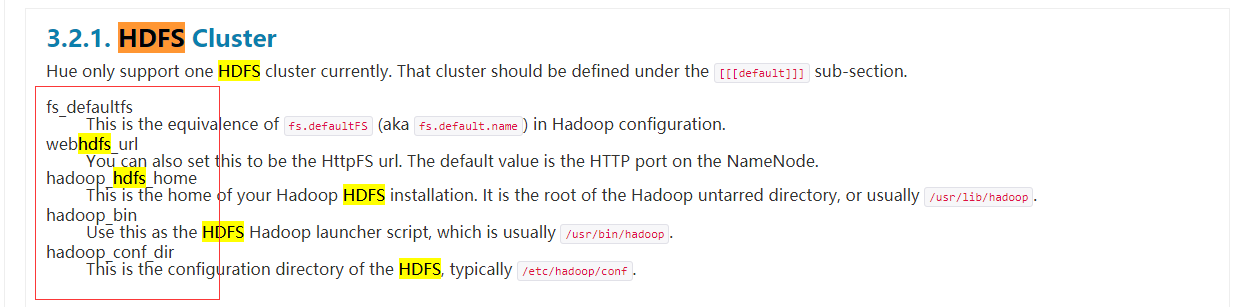
<property>

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

<value>\*</value>

</property>

然后配置hue支持hdfs集群，主要修改一下参数



# vim /root/source/hue-3.9.0-cdh5.13.1/desktop/conf/hue.ini

fs\_defaultfs=hdfs://192.168.75.3:8020

webhdfs\_url=http://192.168.75.3:50070/webhdfs/v1

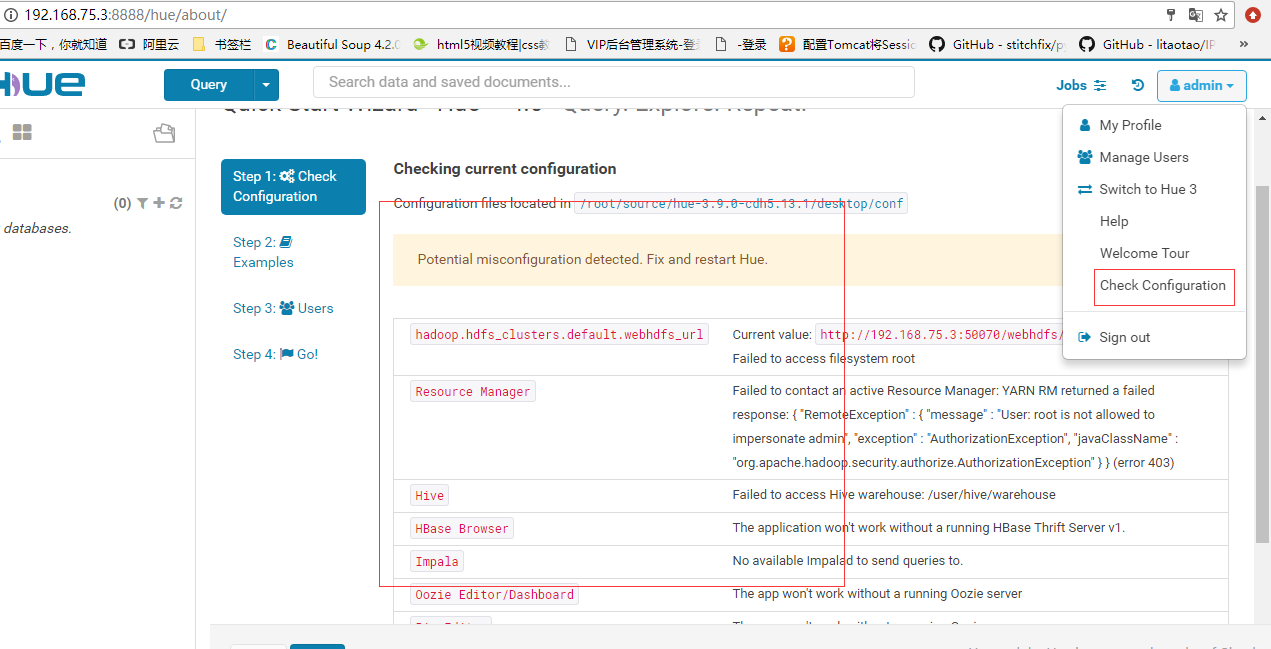
hadoop\_bin=/opt/hadoop-2.7.3

hadoop\_conf\_dir=/opt/hadoop-2.7.3/etc/hadoop

default\_hdfs\_superuser=root

修改完hue的配置文件后，需要重启一下hue才可以正常的使用

访问网页查看一下是否配置有问题：按照如图方式点击查看：



报错一、

Current value: http://192.168.75.3:50070/webhdfs/v1  
Failed to access filesystem root

文件权限有问题：

修改hue的配置文件hue.ini

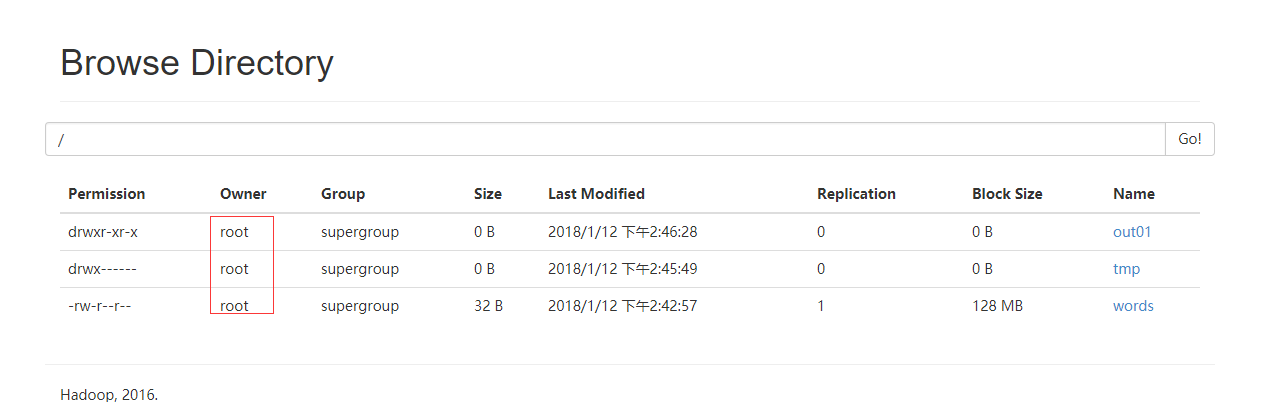
将

##default\_hdfs\_superuser=hdfs

修改为：

default\_hdfs\_superuser=root

重启hue，再次检查错误：



因为我们创建的hdfs的文件都属于root,因此这里也需要使用root权限

配置成功后我们的文件就可以在我们的hue中查看到了：



看到文件以后我们可以进行操作和编辑：

编辑words文件：

但是报错，是权限问题导致的错误，这个时候我们需要修改一下自己的hdfs权限认证：

[root@hadoop01 hadoop-2.7.3]# vim etc/hadoop/hdfs-site.xml

<!-- 关闭hdfs的权限认证-->

<property>

<name>dfs.permissions.enabled</name>

<value>false</value>

</property>

重启一下hdfs:

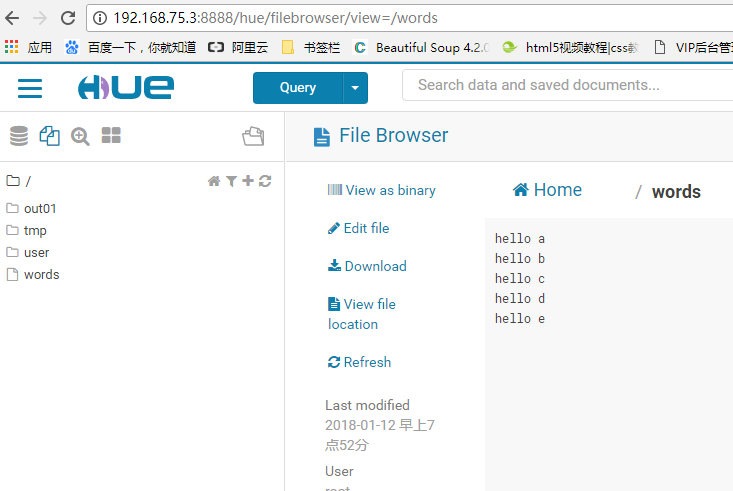
sbin/stop-dfs.sh

sbin/strt-dfs.sh

同时需要在重启一下hue，这个软件不支持热部署，每次都需要重启的：

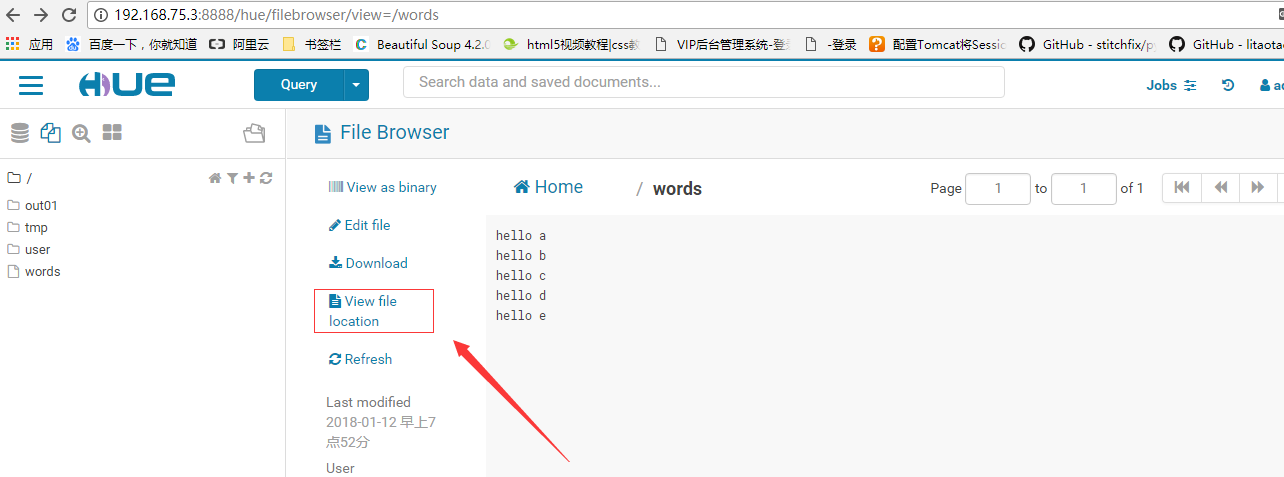
# build/env/bin/supervisor

再次在web页面对我们的文件进行一下修改测试

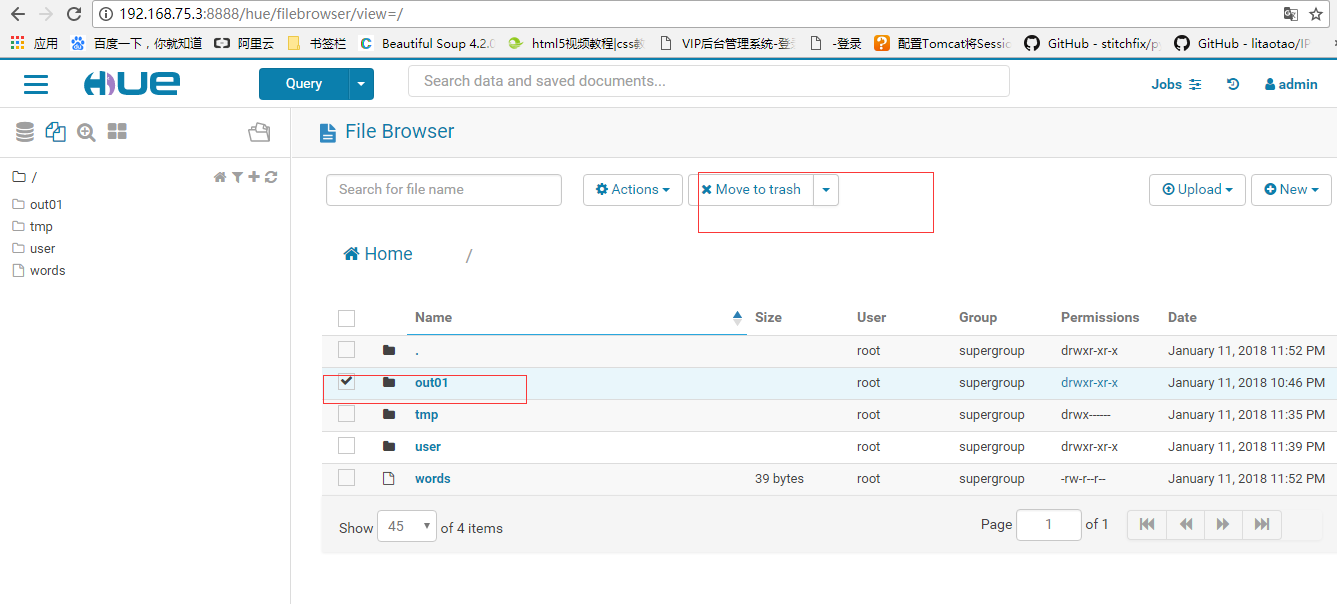


可以成功的操作我们的文件：

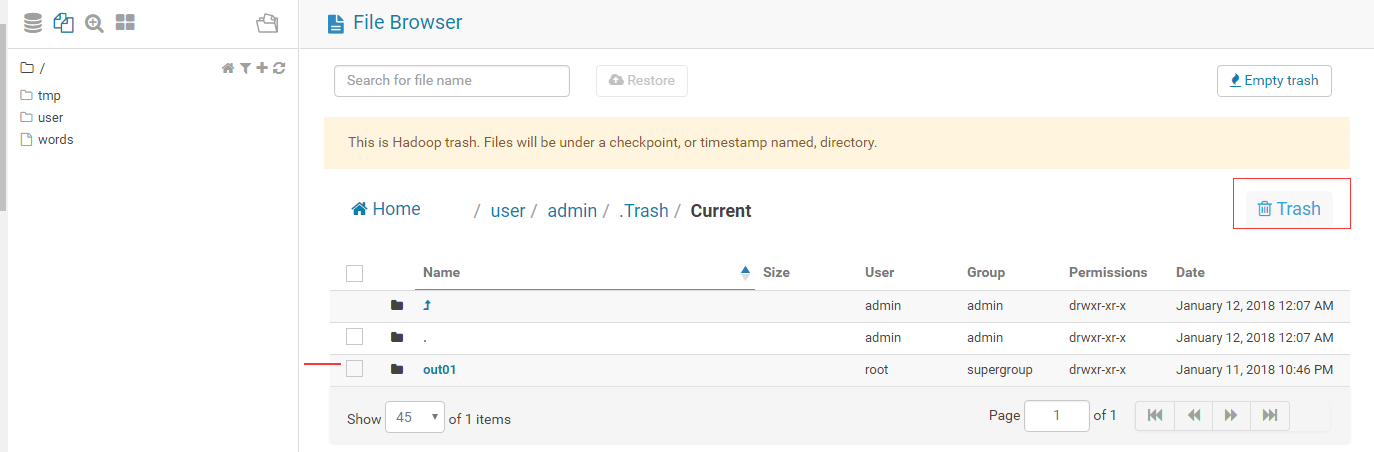
我们还可以删除文件，和进行文件还原：



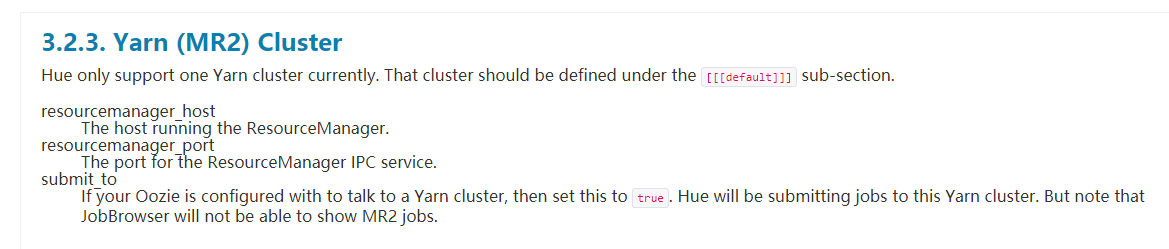
选中文件放入垃圾桶



我们的垃圾桶的current目录中可以找到我们删除的文件或是目录，可以进行还原操作



下一步开始配置yarn,我们的任务节点



参看配置文件，需要修改三个参数

Vim hue

resourcemanager\_host=192.168.75.3

resourcemanager\_port=8032

submit\_to=True：

同时添加一下安全密钥：

secret\_key=jFE93j;2[290-eiw.KEiwN2s3['d;/.q[eIW^y#e=+Iei\*@Mn<qW5o

重启hue

查看配置报错，修改一下：

### Checking current configuration

Configuration files located in /root/source/hue-3.9.0-cdh5.13.1/desktop/conf

Potential misconfiguration detected. Fix and restart Hue.

|  |  |
| --- | --- |
| Resource Manager | Failed to contact an active Resource Manager: YARN RM returned a failed response: { "RemoteException" : { "message" : "User: hue is not allowed to impersonate admin", "exception" : "AuthorizationException", "javaClassName" : "org.apache.hadoop.security.authorize.AuthorizationException" } } (error 403) |

修改：

# vim etc/hadoop/hdfs-site.xml

将

<property>

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

<value>\*</value>

</property>

<property>

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

<value>\*</value>

</property>

修改为

<property>

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

<value>\*</value>

</property>

<property>

<name>hadoop.proxyuser.root.groups</name>

<value>\*</value>

</property>

重启服务：

sbin/stop-dfs.sh

sbin/start-dfs.sh

sbin/stop-yarn.sh

sbin/start-yarn.sh

重启hue

我们在后台启动一个任务：

bin/yarn jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar wordcount /words /out02

在界面中我们可以看到完成状态：

