配置Hue

Step1，下载并解压Hue

https://gethue.com/categories/release/

Step2，配置Hue用户

添加Linux用户 hue，并对相应目录增加权限

adduser hue

chown –R hue:hue /usr/local/hue

Step3，配置Hadoop，在hdfs-site.xml中添加

<property>

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

<value>true</value>

</property>

Step3，启动hadoop

/home/software/Hadoop-2.7.7/sbin/start-dfs.sh

说明etc/hadoop/core-site.xml没有配置正确

vim /home/software/hadoop-2.7.7/etc/hadoop/core-site.xml

添加以下配置

<property>

<name>fs.default.name</name>

<value>hdfs://127.0.0.1:9000</value>

</property>

在core-site.xml中添加

<property>

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

<value>*</value>

</property>

<property>

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

</property>

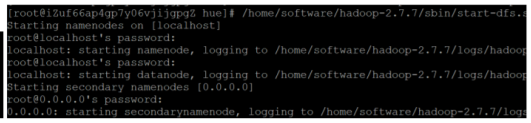
如果你找不到某个文件，可以使用

find / -name hdfs-site.xml

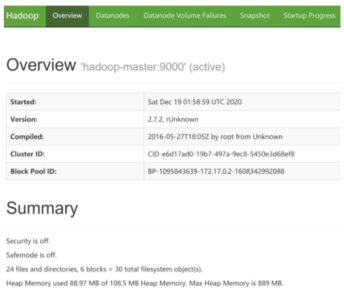
然后使用vim进行编辑

vim /opt/hadoop-hive/config/hdfs-site.xml

配置之后，重新启动Hadoop



然后访问：http://localhost:50070/

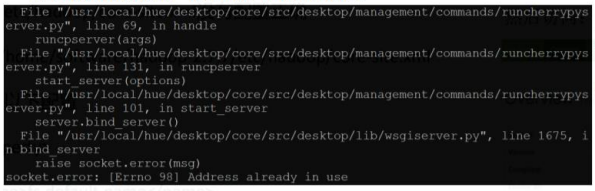


Step4，启动Hue

1）启动Hadoop

2）启动Hue /usr/local/hue/build/env/bin/supervisor

如果启动的时候，报了端口占用的错误



需要查看哪个端口被占用了，然后kill掉

supervisor和hue的进程都需要kill掉

kill -9 17552 17558

2）启动Hue /usr/local/hue/build/env/bin/supervisor

如果启动的时候，报了Table doesn't exist错误

ProgrammingError: (1146, "Table 'hue.desktop_settings' doesn't exist")

```
  File "/usr/local/hue/desktop/core/src/desktop/models.py", line 144, in get_settings
    settings, created = Settings.objects.get_or_create(id=1)
  File "/usr/local/hue/build/env/lib/python2.7/site-packages/Django-1.11.29-py2.7.egg/django/
    return getattr(self.get_queryset(), name)(*args, **kwargs)
  File "/usr/local/hue/build/env/lib/python2.7/site-packages/Django-1.11.29-py2.7.egg/django/
    return self.get(**lookup), False
  File "/usr/local/hue/build/env/lib/python2.7/site-packages/Django-1.11.29-py2.7.egg/django/
    num = len(clone)
  File "/usr/local/hue/build/env/lib/python2.7/site-packages/Django-1.11.29-py2.7.egg/django/
    self._fetch_all()
  File "/usr/local/hue/build/env/lib/python2.7/site-packages/Django-1.11.29-py2.7.egg/django/
    self._result_cache = list(self._iterable_class(self))
  File "/usr/local/hue/build/env/lib/python2.7/site-packages/Django-1.11.29-py2.7.egg/django/
    results = compiler.execute_sql(chunked_fetch=self.chunked_fetch)
  File "/usr/local/hue/build/env/lib/python2.7/site-packages/Django-1.11.29-py2.7.egg/django/
    raise original_exception
ProgrammingError: (1146, "Table 'hue.desktop_settings' doesn't exist")
```

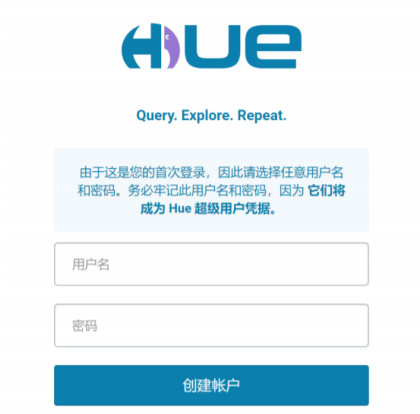
需要初始化数据库

bin/hue syncdb

bin/hue migrate

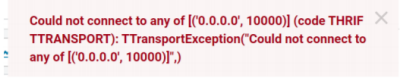
```
[root@i2uf66ap4qp7y06vjijgpgZ env]# bin/hue migrate
System check identified some issues:

WARNINGS:
?: (mysql.W002) MySQL Strict Mode is not set for database connection 'default'
        HINT: MySQL's Strict Mode fixes many data integrity problems in MySQL,
uch as data truncation upon insertion, by escalating warnings into errors. It
 strongly recommended you activate it. See: https://docs.djangoproject.com/en/1
11/ref/databases/#mysql-sql-mode
jobbrowser.DagDetails.dag_info: (fields.W342) Setting unique=True on a ForeignK
y has the same effect as using a OneToOneField.
        HINT: ForeignKey(unique=True) is usually better served by a OneToOneFie
d.
jobbrowser.QueryDetails.hive_query: (fields.W342) Setting unique=True on a Fore
gnKey has the same effect as using a OneToOneField.
        HINT: ForeignKey(unique=True) is usually better served by a OneToOneFie
d.
Operations to perform:
  Apply all migrations: admin, auth, axes, beeswax, contenttypes, desktop, jobs
b, oozie, pig, search, sessions, sites, useradmin
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
```

2）启动Hue /usr/local/hue/build/env/bin/supervisor



首次登录后，创建用户名和密码

登录后，如果Hive没有配置成功，需要重新配置



vim /usr/local/hue/desktop/conf/hue.ini

修改配置如下：

[beeswax]

hive_server_host=node1

hive_server_port=10000

hive_conf_dir=/usr/local/hive/conf

Step1，下载Hive，版本2.3.7

http://hive.apache.org/downloads.html

Step2，解压到/usr/local中

sudo tar -zxvf ./apache-hive-2.3.7-bin.tar.gz -C /usr/local

cd /usr/local/

Step3，将文件夹名改为hive，并修改文件权限

sudo mv apache-hive-2.3.7-bin hive

sudo chown -R hue:hue hive

这里hue:hue为用户组和用户名

Step4，添加环境变量

为了方便，把hive命令加入到环境变量中去，使用vim编辑器打开.bashrc文件：vim ~/.bashrc

添加环境变量：

export HIVE_HOME=/usr/local/hive

export PATH=$PATH:$HIVE_HOME/bin

export HADOOP_HOME=/usr/local/hadoop

使刚配置的环境变量生效：

source ~/.bashrc

Step5，将hive-default.xml.template重命名为hive-default.xml：

cd /usr/local/hive/conf

mv hive-default.xml.template hive-default.xml

Step6，在/usr/local/hive/conf下创建hive-site.xml

使用vim新建配置文件hive-site.xml

cd /usr/local/hive/conf

vim hive-site.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
 <property>
  <name>javax.jdo.option.ConnectionURL</name>
<value>jdbc:mysql://localhost:3306/hive?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>hive</value>
  <description>username to use against metastore
database</description>
 </property>
 <property>
  <name>javax.jdo.option.ConnectionPassword</name>
  <value>hive</value>
  <description>password to use against metastore
database</description>
 </property>
</configuration>
```

hive-site.xml

也可以采用docker安装

- 启动docker

sudo service docker start

- hadoop环境:

http://47.103.118.79:50070/dfshealth.html#tab-overview

- 进入命令：

sudo docker exec -it hadoop-master bash

- 启动hadoop集群（已经启动过了，不要重复启动）

bash start-hadoop.sh

- jps可以查看进程

- 进入hive:

/usr/local/hive/bin/hive

查看数据库：

SHOW DATABASES;

查看所有docker容器

docker ps –a

查看正在运行的容器

docker ps

- 使用HUE

https://demo.gethue.com/hue/accounts/login?next=/