# 开源集群配置Kerberos认证

总体步骤：

1. 生成kerberos认证文件
2. 修改配置文件
3. 重启服务
4. 用户认证
5. 正常提交任务

# 1.kerberos安装及配置

## 1.1 安装 Kerberos

在 kudu1 上安装包 krb5、krb5-server 和 krb5-client。

$ yum install krb5-server -y

在其他节点（kud1、kud2、kud3）安装 krb5-devel、krb5-workstation ：

#使用无密码登陆

$ ssh cdh1 "yum install krb5-devel krb5-workstation -y"

$ ssh cdh2 "yum install krb5-devel krb5-workstation -y"

$ ssh cdh3 "yum install krb5-devel krb5-workstation -y"

## 1.2 修改配置文件

kdc 服务器涉及到三个配置文件：

/etc/krb5.conf

/var/kerberos/krb5kdc/kdc.conf

/var/kerberos/krb5kdc/kadm5.acl

vi /etc/krb5.conf。

[logging]

default = FILE:/var/log/krb5libs.log

kdc = FILE:/var/log/krb5kdc.log

admin\_server = FILE:/var/log/kadmind.log

[libdefaults]

default\_realm = DTSTACK.COM

dns\_lookup\_realm = false

dns\_lookup\_kdc = false

ticket\_lifetime = 24h

renew\_lifetime = 7d

forwardable = true

[realms]

DTSTACK.COM = {

kdc = kudu1:88

admin\_server = kudu1:749

}

[domain\_realm]

.dtstack.com = DTSTACK.COM

dtstack.com = DTSTACK.COM

vi /var/kerberos/krb5kdc/kdc.conf

[kdcdefaults]

kdc\_ports = 88

kdc\_tcp\_ports = 88

[realms]

DTSTACK.COM = {

#master\_key\_type = aes256-cts

max\_renewable\_life = 10d

acl\_file = /var/kerberos/krb5kdc/kadm5.acl

dict\_file = /usr/share/dict/words

admin\_keytab = /var/kerberos/krb5kdc/kadm5.keytab

supported\_enctypes = aes256-cts:normal aes128-cts:normal des3-hmac-sha1:normal arcfour-hmac:normal des-hmac-sha1:normal des-cbc-md5:normal des-cbc-crc:normal

}

vi /var/kerberos/krb5kdc/kadm5.acl

\*/admin@DTSTACK.COM \*

## 1.3 同步配置文件

将 kdc 中的 /etc/krb5.conf 拷贝到集群中其他服务器即可。

$ scp /etc/krb5.conf kudu2:/etc/krb5.conf $ scp /etc/krb5.conf kudu3:/etc/krb5.conf

## 1.4 创建数据库

在 cdh1 上运行初始化数据库命令。其中 -r 指定对应 realm。

$ kdb5\_util create -r DTSTACK.COM -s

## 1.5 启动服务

在 cdh1 节点上运行：

$ chkconfig --level 35 krb5kdc on

$ chkconfig --level 35 kadmin on

$ service krb5kdc start

$ service kadmin start

## 1.6 创建 kerberos 管理员

$ kadmin.local -q "addprinc root/admin"

# 也可以不用手动输入密码

$ echo -e "root\nroot" | kadmin.local -q "addprinc root/admin"

# 或者运行下面命令

$ kadmin.local <<eoj addprinc -pw root root/admin eoj

系统会提示输入密码，密码不能为空，且需妥善保存。

# 2. HDFS 上配置 kerberos

## 2.1 创建认证规则

创建hdfs规则：

kadmin.local -q "addprinc -randkey admin/kudu1@DTSTACK.COM"

kadmin.local -q "addprinc -randkey admin/kudu2@DTSTACK.COM"

kadmin.local -q "addprinc -randkey admin/kudu3@DTSTACK.COM"

kadmin.local -q "addprinc -randkey admin/nn1@DTSTACK.COM"

kadmin.local -q "addprinc -randkey [admin/nn2@DTSTACK.COM"](mailto:hdfs/nn2@DTSTACK.COM\")

创建http规则：

kadmin.local -q "addprinc -randkey HTTP/kudu1@DTSTACK.COM"

kadmin.local -q "addprinc -randkey HTTP/kudu2@DTSTACK.COM"

kadmin.local -q "addprinc -randkey HTTP/kudu3@DTSTACK.COM"

kadmin.local -q "addprinc -randkey HTTP/nn1@DTSTACK.COM"

kadmin.local -q "addprinc -randkey HTTP/nn2@DTSTACK.COM"

## 2.2 创建keytab文件

cd /var/kerberos/krb5kdc/

kadmin.local -q "xst -k admin-unmerged.keytab admin/kudu1@DTSTACK.COM"

kadmin.local -q "xst -k admin-unmerged.keytab admin/kudu2@DTSTACK.COM"

kadmin.local -q "xst -k admin-unmerged.keytab admin/kudu3@DTSTACK.COM"

kadmin.local -q "xst -k HTTP.keytab HTTP/kudu1@DTSTACK.COM"

kadmin.local -q "xst -k HTTP.keytab HTTP/kudu2@DTSTACK.COM"

kadmin.local -q "xst -k HTTP.keytab [HTTP/kudu3@DTSTACK.COM"](mailto:HTTP/kudu3@DTSTACK.COM\")

用ktutil合并成hdfs.keytab文件：

ktutil

ktutil: rkt admin-unmerged.keytab

ktutil: rkt HTTP.keytab

ktutil: wkt admin.keytab

ktutil: exit

## 2.3 部署kerberos keytab文件

拷贝 hdfs.keytab 文件到其他节点的 $HADOOP\_CONF\_DIR 目录

$ cd /var/kerberos/krb5kdc/

$ scp admin.keytab kudu1:$HADOOP\_CONF\_DIR

$ scp admin.keytab kudu2:$HADOOP\_CONF\_DIR

$ scp admin.keytab kudu3:$HADOOP\_CONF\_DIR

并设置权限，分别在 kudu1、kudu2、kudu3 上执行：

ssh kudu1 "chown admin:hadoop $HADOOP\_CONF\_DIR/admin.keytab ;chmod 400 $HADOOP\_CONF\_DIR/admin.keytab"

ssh kudu2 "chown admin:hadoop $HADOOP\_CONF\_DIR/admin.keytab ;chmod 400 $HADOOP\_CONF\_DIR/admin.keytab"

ssh kudu3 "chown admin:hadoop $HADOOP\_CONF\_DIR/admin.keytab ;chmod 400 $HADOOP\_CONF\_DIR/admin.keytab"

## 2.4 修改 hdfs 配置文件

在集群中所有节点的 core-site.xml 文件中添加下面的配置:

**<property>**

**<name>**hadoop.security.authentication**</name>**

**<value>**kerberos**</value>**

**</property>**

**<property>**

**<name>**hadoop.security.authorization**</name>**

**<value>**true**</value>**

**</property>**

在集群中所有节点的 hdfs-site.xml 文件中添加下面的配置：

**<property>**

**<name>**dfs.block.access.token.enable**</name>**

**<value>**true**</value>**

**</property>**

**<property>**

**<name>**dfs.datanode.data.dir.perm**</name>**

**<value>**700**</value>**

**</property>**

**<property>**

**<name>**dfs.namenode.keytab.file**</name>**

**<value>**/etc/hadoop/conf/admin.keytab**</value>**

**</property>**

**<property>**

**<name>**dfs.namenode.kerberos.principal**</name>**

**<value>**admin/\_HOST@DTSTACK.COM**</value>**

**</property>**

**<property>**

**<name>**dfs.namenode.kerberos.https.principal**</name>**

**<value>**HTTP/\_HOST@DTSTACK.COM**</value>**

**</property>**

**<property>**

**<name>**dfs.datanode.address**</name>**

**<value>**0.0.0.0:1004**</value>**

**</property>**

**<property>**

**<name>**dfs.datanode.http.address**</name>**

**<value>**0.0.0.0:1006**</value>**

**</property>**

**<property>**

**<name>**dfs.datanode.keytab.file**</name>**

**<value>**/etc/hadoop/conf/admin.keytab**</value>**

**</property>**

**<property>**

**<name>**dfs.datanode.kerberos.principal**</name>**

**<value>**admin/\_HOST@DTSTACK.COM**</value>**

**</property>**

**<property>**

**<name>**dfs.datanode.kerberos.https.principal**</name>**

**<value>**HTTP/\_HOST@DTSTACK.COM**</value>**

**</property>**

如果想开启 SSL，请添加（本文不对这部分做说明）：

**<property>**

**<name>**dfs.http.policy**</name>**

**<value>**HTTPS\_ONLY**</value>**

**</property>**

如果 HDFS 配置了 QJM HA，则需要添加（另外，你还要在 zookeeper 上配置 kerberos）：

**<property>**

**<name>**dfs.journalnode.keytab.file**</name>**

**<value>**/etc/hadoop/conf/admin.keytab**</value>**

**</property>**

**<property>**

**<name>**dfs.journalnode.kerberos.principal**</name>**

**<value>**admin/\_HOST@DTSTACK.COM**</value>**

**</property>**

**<property>** **<name>**dfs.journalnode.kerberos.internal.spnego.principal**</name>**

**<value>**HTTP/\_HOST@DTSTACK.COM**</value>**

**</property>**

如果配置了 WebHDFS，则添加：

**<property>**

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

**<value>**true**</value>**

**</property>**

**<property>** **<name>**dfs.web.authentication.kerberos.principal**</name>** **<value>**HTTP/\_HOST@DTSTACK.COM**</value>**

**</property>**

**<property>**

**<name>**dfs.web.authentication.kerberos.keytab**</name>** **<value>**/etc/hadoop/conf/admin.keytab**</value>**

**</property>**

在集群中所有节点的 hadoop-env.sh 文件中添加下面的配置

export HADOOP\_SECURE\_DN\_USER=root

## 2.5 启动 NameNode

$ ssh kudu1 "kinit -k -t $HADOOP\_CONF\_DIR/admin.keytab admin/kudu1@DTSTACK.COM "

$ ssh kudu2 "kinit -k -t $HADOOP\_CONF\_DIR/admin.keytab admin/kudu1@DTSTACK.COM "

$ ssh kudu3 "kinit -k -t $HADOOP\_CONF\_DIR/admin.keytab admin/kudu1@DTSTACK.COM "

获取 cdh1的 ticket：

$ kinit -k -t $HADOOP\_CONF\_DIR/admin.keytab admin/kudu1@DTSTACK.COM

如果出现下面异常 kinit: Password incorrect while getting initial credentials，则重新导出 keytab 再试试。

然后启动服务，观察日志：

$ ./start-dfs.sh

$ hadoop fs -ls /

# 3.YARN配置Kerberos认证

## 3.1 生成 keytab

直接使用admin.keytab文件

## 3.2 修改 YARN 配置文件

**<property>**

**<name>**yarn.resourcemanager.keytab**</name>**

**<value>**/etc/hadoop/conf/admin.keytab**</value>**

**</property>**

**<property>**

**<name>**yarn.resourcemanager.principal**</name>**

**<value>**admin/\_HOST@DTSTACK.COM**</value>**

**</property>**

**<property>**

**<name>**yarn.nodemanager.keytab**</name>**

**<value>**/etc/hadoop/conf/admin.keytab**</value>**

**</property>**

**<property>**

**<name>**yarn.nodemanager.principal**</name>**

**<value>**admin/\_HOST@DTSTACK.COM**</value>**

**</property>**

**<property>**

**<name>**yarn.nodemanager.container-executor.class**</name>**

**<value>**org.apache.hadoop.yarn.server.nodemanager.LinuxContainerExecutor**</value>**

**</property>**

**<property>** **<name>**yarn.nodemanager.linux-container-executor.group**</name>** **<value>**admin**</value>**

**</property>**

在 /etc/hadoop/conf 目录下创建 container-executor.cfg 文件，内容如下：

yarn.nodemanager.linux-container-executor.group=admin

banned.users=root

min.user.id=10

allowed.system.users=admin

## 3.3 启动服务

启动 ResourceManager

resourcemanager 是通过 yarn 用户启动的，故在 kudu1 上先获取 yarn 用户的 ticket 再启动服务：

$ kinit -k -t /etc/hadoop/conf/admin.keytab admin/kudu1@DTSTACK.COM

$ ./start-yarn.sh

然后查看日志，确认是否启动成功。

# 4.Hive配置Kerberos认证

## 4.1 生成 keytab

直接使用admin.keytab文件

## 4.2 修改 hive 配置文件

修改 hive-site.xml，添加下面配置：

**<property>**

**<name>**hive.server2.authentication**</name>**

**<value>**KERBEROS**</value>**

**</property>**

**<property>** **<name>**hive.server2.authentication.kerberos.principal**</name>** **<value>**admin/\_HOST@DTSTACK.COM**</value>**

**</property>**

**<property>**

**<name>**hive.server2.authentication.kerberos.keytab**</name>**

**<value>**/etc/hive/conf/admin.keytab**</value>**

**</property>**

**<property>**

**<name>**hive.metastore.sasl.enabled**</name>**

**<value>**true**</value>** **</property>**

**<property>**

**<name>**hive.metastore.kerberos.keytab.file**</name>**

**<value>**/etc/hive/conf/admin.keytab**</value>**

**</property>**

**<property>**

**<name>**hive.metastore.kerberos.principal**</name>**

**<value>**admin/\_HOST@DTSTACK.COM**</value>**

**</property>**

在 core-site.xml 中添加：

**<property>**

**<name>**hadoop.proxyuser.admin.hosts**</name>**

**<value>**\***</value>**

**</property>**

**<property>**

**<name>**hadoop.proxyuser.admin.groups**</name>**

**<value>**\***</value>**

**</property>**

## 4.3 启动服务

启动 Hive MetaStore

hive-metastore 是通过 hive 用户启动的，故在 kudu1 上先获取 hive 用户的 ticket 再启动服务：

$ kinit -k -t /etc/hive/conf/admin.keytab admin/kudu1@DTSTACK.COM

$ ./metastore.sh start

然后查看日志，确认是否启动成功。

启动 Hive Server2

hive-server2 是通过 hive 用户启动的，故在 kudu2 和 kudu3 上先获取 hive 用户的 ticket 再启动服务：

$ kinit -k -t /etc/hive/conf/admin.keytab admin/kudu1@DTSTACK.COM

$ ./hive-server2.sh start

然后查看日志，确认是否启动成功。

# 5. Zookeeper配置Kerberos认证

## 5.1 生成 keytab

在 cdh1 节点，即 KDC server 节点上执行下面命令：

$ cd /var/kerberos/krb5kdc/

kadmin.local -q "addprinc -randkey zookeeper/kudu1@DTSTACK.COM "

kadmin.local -q "addprinc -randkey zookeeper/kudu2@DTSTACK.COM "

kadmin.local -q "addprinc -randkey zookeeper/kudu3@DTSTACK.COM "

kadmin.local -q "xst -k zookeeper.keytab zookeeper/kudu1@DTSTACK.COM " kadmin.local -q "xst -k zookeeper.keytab zookeeper/kudu2@DTSTACK.COM " kadmin.local -q "xst -k zookeeper.keytab zookeeper/kudu3@DTSTACK.COM "

拷贝 zookeeper.keytab 文件到其他节点的 /etc/zookeeper/conf 目录：

$ scp zookeeper.keytab kudu1:/opt/dtstack/zookeeper/conf

$ scp zookeeper.keytab kudu2:/opt/dtstack/zookeeper/conf

$ scp zookeeper.keytab kudu3:/opt/dtstack/zookeeper/conf

client端使用之前生成的admin.keytab文件

## 5.2 修改 zookeeper 配置文件

在 cdh1 节点上修改 /opt/dtstack/zookeeper/conf/zoo.cfg 文件，添加下面内容：

authProvider.1=org.apache.zookeeper.server.auth.SASLAuthenticationProvider jaasLoginRenew=3600000

将修改的上面文件同步到其他节点：cdh2、cdh3：

$ scp /opt/dtstack/zookeeper/conf/zoo.cfg kudu2:/opt/dtstack/zookeeper/conf/zoo.cfg

$ scp /opt/dtstack/zookeeper/conf/zoo.cfg kudu3:/opt/dtstack/zookeeper/conf/zoo.cfg

## 5.3 创建 JAAS 配置文件

在 cdh1 的配置文件目录创建 jaas.conf 文件，内容如下：

Server {

com.sun.security.auth.module.Krb5LoginModule required

useKeyTab=true

keyTab="/opt/dtstack/zookeeper/conf/zookeeper.keytab"

storeKey=true

useTicketCache=false

principal="zookeeper/kudu1@DTSTACK.COM";

};

Client {

com.sun.security.auth.module.Krb5LoginModule required

useKeyTab=true

keyTab="/opt/dtstack/hadoop/conf/admin.keytab"

storeKey=true

useTicketCache=false

principal="admin/kudu1@DTSTACK.COM";

};

同样，在 cdh2 和 cdh3 节点也创建该文件，注意每个节点的 principal 有所不同。

然后，在 /opt/dtstack/zookeeper/conf/ 目录创建 java.env，内容如下：

export JVMFLAGS="-Djava.security.auth.login.config=/opt/dtstack/zookeeper/conf/jaas.conf"

并将该文件同步到其他节点：

$ scp /opt/dtstack/zookeeper/conf/java.env kudu2:/opt/dtstack/zookeeper/conf/java.env

$ scp /opt/dtstack/zookeeper/conf/java.env kudu3:/opt/dtstack/zookeeper/conf/java.env

## 5.4 重启服务

依次重启，并观察日志：

/opt/dtstack/zookeeper/bin/zkEnv.sh restart