Hive配置Kerberos认证

2014.11.06 | Comments

关于 Kerberos 的安装和 HDFS 配置 kerberos 认证，请参考 HDFS配置kerberos认证(/2014/11/04/config-kerberos-in-cdh-hdfs.html)。

关于 Kerberos 的安装和 YARN 配置 kerberos 认证，请参考 YARN配置kerberos认证(/2014/11/05/config-kerberos-in-cdh-yarn.html)。

1. 环境说明

系统环境：

- 操作系统：CentOs 6.6
- Hadoop版本： CDH5.4
- JDK版本： 1.7.0_71
- 运行用户：root

集群各节点角色规划为：

```
192.168.56.121        cdh1     NameNode、ResourceManager、HBase、Hive metastore、Impala Catalo
g、Impala statestore、Sentry
192.168.56.122        cdh2     DataNode、SecondaryNameNode、NodeManager、HBase、Hive Server2、
Impala Server
192.168.56.123        cdh3     DataNode、HBase、NodeManager、Hive Server2、Impala Server
```

cdh1作为master节点，其他节点作为slave节点，hostname 请使用小写，要不然在集成 kerberos 时会出现一些错误。

2. 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 hive/cdh1@JAVACHEN.COM "
kadmin.local -q "addprinc -randkey hive/cdh2@JAVACHEN.COM "
kadmin.local -q "addprinc -randkey hive/cdh3@JAVACHEN.COM "

kadmin.local -q "xst  -k hive.keytab  hive/cdh1@JAVACHEN.COM "
kadmin.local -q "xst  -k hive.keytab  hive/cdh2@JAVACHEN.COM "
kadmin.local -q "xst  -k hive.keytab  hive/cdh3@JAVACHEN.COM "
```

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

```
$ scp hive.keytab cdh1:/etc/hive/conf
$ scp hive.keytab cdh2:/etc/hive/conf
$ scp hive.keytab cdh3:/etc/hive/conf
```

并设置权限，分别在 cdh1、cdh2、cdh3 上执行：

```
$ ssh cdh1 "cd /etc/hive/conf/;chown hive:hadoop hive.keytab ;chmod 400 *.keytab"
$ ssh cdh2 "cd /etc/hive/conf/;chown hive:hadoop hive.keytab ;chmod 400 *.keytab"
$ ssh cdh3 "cd /etc/hive/conf/;chown hive:hadoop hive.keytab ;chmod 400 *.keytab"
```

由于 keytab 相当于有了永久凭证，不需要提供密码(如果修改 kdc 中的 principal 的密码，则该 keytab 就会失效)，所以其他用户如果对该文件有读权限，就可以冒充 keytab 中指定的用户身份访问 hadoop，所以 keytab 文件需要确保只对 owner 有读权限(0400)

3. 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>hive/_HOST@JAVACHEN.COM</value>
</property>
<property>
  <name>hive.server2.authentication.kerberos.keytab</name>
  <value>/etc/hive/conf/hive.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/hive.keytab</value>
</property>
<property>
  <name>hive.metastore.kerberos.principal</name>
  <value>hive/_HOST@JAVACHEN.COM</value>
</property>
```

在 core-site.xml 中添加：

```
<property>
  <name>hadoop.proxyuser.hive.hosts</name>
  <value>*</value>
</property>
<property>
  <name>hadoop.proxyuser.hive.groups</name>
  <value>*</value>
</property>
<property>
  <name>hadoop.proxyuser.hdfs.hosts</name>
  <value>*</value>
</property>
<property>
  <name>hadoop.proxyuser.hdfs.groups</name>
  <value>*</value>
</property>
<property>
  <name>hadoop.proxyuser.HTTP.hosts</name>
  <value>*</value>
</property>
<property>
  <name>hadoop.proxyuser.HTTP.groups</name>
  <value>*</value>
</property>
```

记住将修改的上面文件同步到其他节点：cdh2、cdh3，并再次一一检查权限是否正确。

```
$ scp /etc/hive/conf/hive-site.xml cdh2:/etc/hive/conf/
$ scp /etc/hive/conf/hive-site.xml cdh3:/etc/hive/conf/
```

4．启动服务

启动 Hive MetaStore

hive-metastore 是通过 hive 用户启动的，故在 cdh1 上先获取 hive 用户的 ticket 再启动服务：

```
$ kinit -k -t /etc/hive/conf/hive.keytab hive/cdh1@JAVACHEN.COM
$ service hive-metastore 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 cdh2 和 cdh3 上先获取 hive 用户的 ticket 再启动服务：

```
$ kinit -k -t /etc/hive/conf/hive.keytab hive/cdh1@JAVACHEN.COM
$ service hive-server2 start
```

然后查看日志，确认是否启动成功。

5．测试

Hive CLI

在没有配置 kerberos 之前，想要通过 hive 用户运行 hive 命令需要执行sudo，现在配置了 kerberos 之后，不再需要 `sudo` 了，hive 会通过 ticket 中的用户去执行该命令：

```
$ klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: hdfs/dn5.h.lashou-inc.com@lashou_hadoop

Valid starting     Expires            Service principal
11/06/14 11:39:09  11/07/14 11:39:09  krbtgt/lashou_hadoop@lashou_hadoop
  renew until 11/08/14 11:39:09


Kerberos 4 ticket cache: /tmp/tkt0
klist: You have no tickets cached
```

运行Hive cli：

```
$ hive
hive> set system:user.name;
system:user.name=root
hive> create table t(id int);
OK
Time taken: 2.183 seconds
hive> show tables;
OK
t
Time taken: 1.349 seconds
hive> select * from t;
OK
Time taken: 1.116 seconds
```

可以看到在获取了 hdfs 用户的 ticket 之后，进入 hive cli 可以执行查看表、查询数据等命令。当然，你也可以获取 hive 的 ticket 之后再来运行 hive 命令。

另外，如果你想通过普通用户来访问 hive，则需要 kerberos 创建规则和导出 ticket，然后把这个 ticket 拷贝到普通用户所在的家目录，在获取 ticket 了之后，再运行 hive 命令即可。

JDBC 客户端

客户端通过 jdbc 代码连结 hive-server2：

```
String url = "jdbc:hive2://cdh1:10000/default;principal=hive/cdh1@JAVACHEN.COM"
Connection con = DriverManager.getConnection(url);
```

Beeline

Beeline 连结 hive-server2：

```
$ beeline
beeline> !connect jdbc:hive2://cdh1:10000/default;principal=hive/cdh1@JAVACHEN.COM
scan complete in 4ms
Connecting to jdbc:hive2://localhost:10000/default;principal=hive/cdh1@JAVACHEN.COM;
Enter username for jdbc:hive2://localhost:10000/default;principal=hive/cdh1@JAVACHEN.COM;:
Enter password for jdbc:hive2://localhost:10000/default;principal=hive/cdh1@JAVACHEN.COM;:
Connected to: Apache Hive (version 0.14.0)
Driver: Hive (version 0.14.0-cdh5.4.0)
Transaction isolation: TRANSACTION_REPEATABLE_READ
0: jdbc:hive2://cdh1:10000/default> select * from t;
+-------+--+
| t.id  |
+-------+--+
+-------+--+
No rows selected (1.575 seconds)
0: jdbc:hive2://cdh1:10000/default> desc t;
+-----------+------------+----------+--+
| col_name  | data_type  | comment  |
+-----------+------------+----------+--+
| id        | int        |          |
+-----------+------------+----------+--+
1 row selected (0.24 seconds)
```