# Hadoop配置LDAP集成Kerberos

2014.11.12 | Comments

本文主要记录 cdh hadoop 集群集成 ldap 的过程，这里 ldap 安装的是 OpenLDAP (http://www.openldap.org/) 。LDAP 用来做账号管理，Kerberos作为认证。授权一般来说是由应用来决定的，通过在 LDAP 数据库中配置一些属性可以让应用程序来进行授权判断。

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)。

## 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
- OpenLDAP 版本：2.4.39
- Kerberos 版本：1.10.3
- 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 Catalog、Impala statestore、Sentry
192.168.56.122        cdh2     DataNode、NodeManager、HBase、Hiveserver2、Impala Server
192.168.56.123        cdh3     DataNode、HBase、NodeManager、Hiveserver2、Impala Server
```

cdh1作为master节点，其他节点作为slave节点，我们在cdh1节点安装kerberos Server，在其他节点安装kerberos client。

## 2．安装服务端

### 2.1 安装

同安装 kerberos 一样，这里使用 cdh1 作为服务端安装 openldap。

```
$ yum install db4 db4-utils db4-devel cyrus-sasl* krb5-server-ldap -y
$ yum install openldap openldap-servers openldap-clients openldap-devel compat-openldap -y
```

查看安装的版本：

```
$ rpm -qa openldap
openldap-2.4.39-8.el6.x86_64

$ rpm -qa krb5-server-ldap
krb5-server-ldap-1.10.3-33.el6.x86_64
```

### 2.2 OpenSSL

> 如果，你不配置ssl，这部分内容可以略过，实际安装过程中，我也没有详细去操作这部分内容。

OpenLDAP 默认使用 Mozilla NSS，安装后已经生成了一份证书，可使用 `certutil -d /etc/openldap/certs/ -L -n 'OpenLDAP Server'` 命令查看。使用如下命令生成RFC格式CA证书并分发到客户机待用。

```
$ certutil -d /etc/openldap/certs/ -L -a -n 'OpenLDAP Server' -f /etc/openldap/certs/password > /etc/openldap/ldapCA.rfc

# 拷贝到其他节点
$ scp /etc/openldap/ldapCA.rfc cdh2:/tmp
$ scp /etc/openldap/ldapCA.rfc cdh3:/tmp
```

附，生成自签名证书的命令供参考：

```
$ certutil -d /etc/openldap/certs -S -n 'test cert' -x -t 'u,u,u' -s 'C=XX, ST=Default Province, L=Default City, O=Default Company Ltd, OU=Defa
ult Unit, CN=cdh1' -k rsa -v 120 -f /etc/openldap/certs/password
```

修改 `/etc/sysconfig/ldap`，开启 ldaps：

```
# Run slapd with -h "... ldaps:/// ..."
#   yes/no, default: no
SLAPD_LDAPS=yes
```

### 2.3 LDAP 服务端配置

### 2.3 LDAP 服务端配置

更新配置库：

```
rm -rf /var/lib/ldap/*
cp /usr/share/openldap-servers/DB_CONFIG.example /var/lib/ldap/DB_CONFIG
chown -R ldap.ldap /var/lib/ldap
```

在2.4以前的版本中，OpenLDAP 使用 slapd.conf 配置文件来进行服务器的配置，而2.4开始则使用 `slapd.d` 目录保存细分后的各种配置，这一点需要注意，其数据存储位置即目录 `/etc/openldap/slapd.d` 。尽管该系统的数据文件是透明格式的，还是建议使用 ldapadd，ldapdelete，ldapmodify 等命令来修改而不是直接编辑。

默认配置文件保存在 /etc/openldap/slapd.d，将其备份：

```
cp -rf /etc/openldap/slapd.d /etc/openldap/slapd.d.bak
```

添加一些基本配置，并引入 kerberos 和 openldap 的 schema：

```
$ cp /usr/share/doc/krb5-server-ldap-1.10.3/kerberos.schema /etc/openldap/schema/

$ touch /etc/openldap/slapd.conf

$ echo "include /etc/openldap/schema/corba.schema
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/duaconf.schema
include /etc/openldap/schema/dyngroup.schema
include /etc/openldap/schema/inetorgperson.schema
include /etc/openldap/schema/java.schema
include /etc/openldap/schema/misc.schema
include /etc/openldap/schema/nis.schema
include /etc/openldap/schema/openldap.schema
include /etc/openldap/schema/ppolicy.schema
include /etc/openldap/schema/collective.schema
include /etc/openldap/schema/kerberos.schema" > /etc/openldap/slapd.conf
$ echo -e "pidfile /var/run/openldap/slapd.pid\nargsfile /var/run/openldap/slapd.args" >> /etc/openldap/slapd.conf

#更新slapd.d
$ slaptest -f /etc/openldap/slapd.conf -F /etc/openldap/slapd.d

$ chown -R ldap:ldap /etc/openldap/slapd.d && chmod -R 700 /etc/openldap/slapd.d
```

### 2.4 启动服务

启动 LDAP 服务：

```
chkconfig --add slapd
chkconfig --level 345 slapd on

/etc/init.d/slapd start
```

查看状态，验证服务端口：

```
$ ps aux | grep slapd | grep -v grep
  ldap      9225  0.0  0.2 581188 44576 ?        Ssl  15:13   0:00 /usr/sbin/slapd -h ldap:/// -u ldap

$ netstat -tunlp  | grep :389
  tcp        0      0 0.0.0.0:389                 0.0.0.0:*                   LISTEN      8510/slapd
  tcp        0      0 :::389                      :::*                        LISTEN      8510/slapd
```

如果启动失败，则运行下面命令来启动 slapd 服务并查看日志：

```
$ slapd -h ldap://127.0.0.1 -d 481
```

待查明原因之后，停止该进程使用正常方式启动 slapd 服务。

### 2.5 LDAP 和 Kerberos

在Kerberos安全机制里，一个principal就是realm里的一个对象，一个principal总是和一个密钥（secret key）成对出现的。

这个principal的对应物可以是service，可以是host，也可以是user，对于Kerberos来说，都没有区别。

Kdc(Key distribute center)知道所有principal的secret key，但每个principal对应的对象只知道自己的那个secret key。这也是“共享密钥”的由来。

为了使 Kerberos 能够绑定到 OpenLDAP 服务器，请创建一个管理员用户和一个 principal，并生成 keytab 文件，设置该文件的权限为 LDAP 服务运行用户可读（ LDAP 服务运行用户一般为 ldap）：

```
$ kadmin.local -q "addprinc ldapadmin@JAVACHEN.COM"
$ kadmin.local -q "addprinc -randkey ldap/cdh1@JAVACHEN.COM"
$ kadmin.local -q "ktadd -k /etc/openldap/ldap.keytab ldap/cdh1@JAVACHEN.COM"

$ chown ldap:ldap /etc/openldap/ldap.keytab && chmod 640 /etc/openldap/ldap.keytab
```

ktadd 后面的 `-k` 指定把 key 存放在一个本地文件中。

使用 ldapadmin 用户测试：

```
kinit ldapadmin
```

系统会提示输入密码，如果一切正常，那么会安静的返回。实际上，你已经通过了kerberos的身份验证，且获得了一个Service TGT(Ticket-Granting Ticket). Service TGT的意义是， 在一段时间内，你都可以用此TGT去请求某些service，比如ldap service，而不需要再次通过kerberos的认证。

确保 LDAP 启动时使用上一步中创建的keytab文件，在 `/etc/sysconfig/ldap` 增加 `KRB5_KTNAME` 配置：

```
export KRB5_KTNAME=/etc/openldap/ldap.keytab
```

然后，重启 slapd 服务。

## 2.6 创建数据库

进入到 /etc/openldap/slapd.d 目录，查看 `etc/openldap/slapd.d/cn\=config/olcDatabase={2}bdb.ldif` 可以看到一些默认的配置，例如：

```
olcRootDN: cn=Manager,dc=my-domain,dc=com
olcRootPW: secret
olcSuffix: dc=my-domain,dc=com
```

接下来更新这三个配置，建立 modify.ldif 文件，内容如下：

```
dn: olcDatabase={2}bdb,cn=config
changetype: modify
replace: olcSuffix
olcSuffix: dc=javachen,dc=com

dn: olcDatabase={2}bdb,cn=config
changetype: modify
replace: olcRootDN
# Temporary lines to allow initial setup
olcRootDN: uid=ldapadmin,ou=people,dc=javachen,dc=com

dn: olcDatabase={2}bdb,cn=config
changetype: modify
add: olcRootPW
olcRootPW: secret

dn: cn=config
changetype: modify
add: olcAuthzRegexp
olcAuthzRegexp: uid=([^,]*),cn=GSSAPI,cn=auth uid=$1,ou=people,dc=javachen,dc=com

dn: olcDatabase={2}bdb,cn=config
changetype: modify
add: olcAccess
# Everyone can read everything
olcAccess: {0}to dn.base="" by * read
# The ldapadm dn has full write access
olcAccess: {1}to * by dn="uid=ldapadmin,ou=people,dc=javachen,dc=com" write by * read
```

说明：

- 上面的密码使用的是明文密码 secret ，你也可以使用 `slappasswd -s secret` 生成的字符串作为密码。
- 上面的权限中指明了只有用户 `uid=ldapadmin,ou=people,dc=javachen,dc=com` 有写权限。

使用下面命令导入更新配置：

```
$ ldapmodify -Y EXTERNAL -H ldapi:/// -f modify.ldif
```

这时候数据库没有数据，需要添加数据，你可以手动编写 ldif 文件来导入一些用户和组，或者使用 migrationtools 工具来生成 ldif 模板。创建 setup.ldif 文件如下：

```
dn: dc=javachen,dc=com
```

```
dn: dc=javachen,dc=com
objectClass: top
objectClass: dcObject
objectclass: organization
o: javachen com
dc: javachen

dn: ou=people,dc=javachen,dc=com
objectclass: organizationalUnit
ou: people
description: Users

dn: ou=group,dc=javachen,dc=com
objectClass: organizationalUnit
ou: group

dn: uid=ldapadmin,ou=people,dc=javachen,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
cn: LDAP admin account
uid: ldapadmin
sn: ldapadmin
uidNumber: 1001
gidNumber: 100
homeDirectory: /home/ldap
loginShell: /bin/bash
```

使用下面命令导入数据，密码是前面设置的 `secret` 。

```
$ ldapadd -x -D "uid=ldapadmin,ou=people,dc=javachen,dc=com" -w secret -f setup.ldif
```

参数说明：

- `-w` 指定密码
- `-x` 是使用一个匿名的绑定

## 2.7 LDAP 的使用

### 导入系统用户

接下来你可以从 `/etc/passwd`, `/etc/shadow`, `/etc/groups` 中生成 `ldif` 更新 `ldap` 数据库，这需要用到 `migrationtools` 工具。

安装：

```
$ yum install migrationtools -y
```

利用迁移工具生成模板，先修改默认的配置：

```
$ vim /usr/share/migrationtools/migrate_common.ph

#line 71 defalut DNS domain
$DEFAULT_MAIL_DOMAIN = "javachen.com";
#line 74 defalut base
$DEFAULT_BASE = "dc=javachen,dc=com";
```

生成模板文件：

```
/usr/share/migrationtools/migrate_base.pl > /opt/base.ldif
```

然后，可以修改该文件，然后执行导入命令：

```
$ ldapadd -x -D "uid=ldapadmin,ou=people,dc=javachen,dc=com" -w secret -f /opt/base.ldif
```

将当前节点上的用户导入到 `ldap` 中，可以有选择的导入指定的用户：

```
# 先添加用户
$ useradd test hive
# 查找系统上的 test、hive 等用户
$ grep -E "test|hive" /etc/passwd  >/opt/passwd.txt
$ /usr/share/migrationtools/migrate_passwd.pl /opt/passwd.txt /opt/passwd.ldif
$ ldapadd -x -D "uid=ldapadmin,ou=people,dc=javachen,dc=com" -w secret -f /opt/passwd.ldif
```

将用户组导入到 `ldap` 中：

```
# 生成用户组的 ldif 文件，然后导入到 ldap
$ grep -E "test|hive" /etc/group  >/opt/group.txt
$ /usr/share/migrationtools/migrate_group.pl /opt/group.txt /opt/group.ldif
$ ldapadd -x -D "uid=ldapadmin,ou=people,dc=javachen,dc=com" -w secret -f /opt/group.ldif
```

### 查询

查询新添加的 test 用户：

```
$ ldapsearch -LLL -x -D 'uid=ldapadmin,ou=people,dc=javachen,dc=com' -w secret -b 'dc=javachen,dc=com' 'uid=test'
  dn: uid=test,ou=people,dc=javachen,dc=com
  objectClass: inetOrgPerson
  objectClass: posixAccount
  objectClass: shadowAccount
  cn: test account
  sn: test
  uid: test
  uidNumber: 1001
  gidNumber: 100
  homeDirectory: /home/test
  loginShell: /bin/bash
```

可以看到，通过指定‘uid=test’，我们只查询这个用户的数据，这个查询条件叫做filter。有关 filter 的使用可以查看 ldapsearch 的 manpage。

### 修改

用户添加好以后，需要给其设定初始密码，运行命令如下：

```
$ ldappasswd -x -D 'uid=ldapadmin,ou=people,dc=javachen,dc=com' -w secret "uid=test,ou=people,dc=javachen,dc=com" -S
```

### 删除

删除用户或组条目：

```
$ ldapdelete -x -w secret -D 'uid=ldapadmin,ou=people,dc=javachen,dc=com' "uid=test,ou=people,dc=javachen,dc=com"
$ ldapdelete -x -w secret -D 'uid=ldapadmin,ou=people,dc=javachen,dc=com' "cn=test,ou=group,dc=javachen,dc=com"
```

## 3．客户端配置

在 cdh2 和 cdh3上，使用下面命令安装openldap客户端

```
$ yum install openldap-clients -y
```

修改 /etc/openldap/ldap.conf 以下两个配置

```
BASE    dc=javachen,dc=com
URI     ldap://cdh1
```

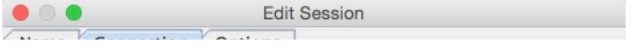
然后，运行下面命令测试：

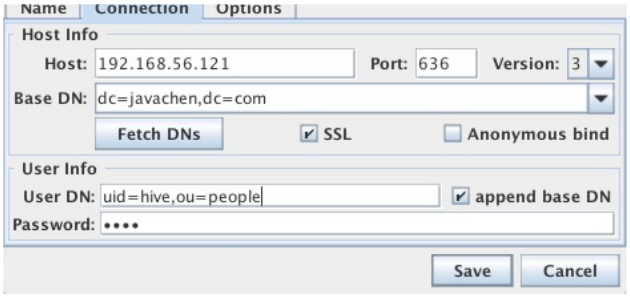
```
#先删除 ticket
$ kdestroy

$ ldapsearch -b 'dc=javachen,dc=com'
  SASL/GSSAPI authentication started
  ldap_sasl_interactive_bind_s: Local error (-2)
    additional info: SASL(-1): generic failure: GSSAPI Error: Unspecified GSS failure.  Minor code may provide more information (No credentials
 cache found)
```

重新获取 ticket：

```
$ kinit root/admin
$ ldapsearch -b 'dc=javachen,dc=com'
 # 没有报错了
$ ldapwhoami
  SASL/GSSAPI authentication started
  SASL username: root/admin@JAVACHEN.COM
  SASL SSF: 56
  SASL installing layers
  dn:uid=root/admin,ou=people,dc=javachen,dc=com
  Result: Success (0)

# 直接输入 ldapsearch 不会报错
$ ldapsearch
```

使用 LDAP 客户端工具进行测试，这里我使用的是 LDAP Browser/Editor：






# 4. 配置 Hive 集成 LDAP

> 说明： CDH5.2 之前 hive-server2 支不支持集成 ldap，故需要升级 cdh 版本到高版本，如 cdh5.3，该版本支持 ldap。

## 修改配置文件

这部分内容参考自Using LDAP Username/Password Authentication with HiveServer2 (http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/cdh_sg_hiveserver2_security.html#topic_9_1_3_unique_1)。

我这使用的是 OpenLDAP ，故修改 hive-site.xml 配置文件如下：

```
<property>
  <name>hive.server2.authentication</name>
  <value>LDAP</value>
</property>
<property>
  <name>hive.server2.authentication.ldap.url</name>
  <value>ldap://cdh1</value>
</property>
<property>
  <name>hive.server2.authentication.ldap.baseDN</name>
  <value>ou=people,dc=javachen,dc=com</value>
</property>
```

为什么这样配置，可以参考 LdapAuthenticationProviderImpl.java (https://svn.apache.org/repos/asf/hive/trunk/service/src/java/org/apache/hive/service/auth/LdapAuthenticationProviderImpl.java) 源码。

## 测试

重启服务：

```
/etc/init.d/hive-server2 restart
```

然后使用 beeline 测试：

```
beeline --verbose=true
beeline> !connect jdbc:hive2://cdh1:10000/default
Connecting to jdbc:hive2://cdh1:10000/default;
Enter username for jdbc:hive2://cdh1:10000/default;: hive
Enter password for jdbc:hive2://cdh1:10000/default;: ****
```

# 5. 配置 Impala 集成 LDAP

## 修改配置文件

修改 /etc/default/impala 中的 `IMPALA_SERVER_ARGS` 参数，添加

```
-enable_ldap_auth=true \
-ldap_uri=ldaps://cdh1 \
-ldap_baseDN=ou=people,dc=javachen,dc=com
```

注意：

- 如果没有开启 ssl，则添加 -ldap_passwords_in_clear_ok=true ，同样如果开启了 ssl，则 ldap_uri 值为 ldaps://XXXX
- ldap_baseDN 的值是 ou=people,dc=javachen,dc=com ，因为 impala 会将其追加到 uid={用户名}, 后面

## 测试

重启服务：

```
$ /etc/init.d/impala-server restart
```

然后使用 impala-shell 测试：

```
$ impala-shell -l -u test
  Starting Impala Shell using LDAP-based authentication
  LDAP password for test:
  Connected to cdh1:21000
  Server version: impalad version 2.0.0-cdh5 RELEASE (build ecf30af0b4d6e56ea80297df2189367ada6b7da7)
  Welcome to the Impala shell. Press TAB twice to see a list of available commands.

  Copyright (c) 2012 Cloudera, Inc. All rights reserved.

  (Shell build version: Impala Shell v2.0.0-cdh5 (ecf30af) built on Sat Oct 11 13:56:06 PDT 2014)
  [cdh1:21000] >
```

使用 beeline 通过 ldap 方式来连接 jdbc 进行测试：

```
$ beeline -u "jdbc:hive2://cdh1:21050/default;" -n test -p test
  scan complete in 2ms
  Connecting to jdbc:hive2://cdh1:21050/default;
  Connected to: Impala (version 2.0.0-cdh5)
  Driver: Hive JDBC (version 0.13.1-cdh5.2.0)
  Transaction isolation: TRANSACTION_REPEATABLE_READ
  Beeline version 0.13.1-cdh5.2.0 by Apache Hive

  0: jdbc:hive2://cdh1:21050/default>show tables;
  +----------------------------+--+
  |            name            |
  +----------------------------+--+
  | t1                         |
  | tab1                       |
  | tab2                       |
  | tab3                       |
  +----------------------------+--+
  4 rows selected (0.325 seconds)
```

## 6．参考文章


- New in CDH 5.2: Impala Authentication with LDAP and Kerberos (http://www.tuicool.com/articles/6fy6z2r)
- 使用 LDAP + Kerberos 实现集中用户认证及授权系统 (http://blog.clanzx.net/2013/09/27/ldap-kerberos.html)
- Linux NFS服务器的安装与配置 (http://www.cnblogs.com/mchina/archive/2013/01/03/2840040.html)
- linux的LDAP认证服务器的配置及客户端pam网络验证实例 (http://blog.csdn.net/kakane/article/details/7455922)
- Kerberos and LDAP (https://help.ubuntu.com/10.04/serverguide/kerberos-ldap.html)
- RHEL6配置简单LDAP服务器 (http://blog.sina.com.cn/s/blog_64aac6750101gwst.html)
- 使用 LDAP 和 Kerberos (https://www.suse.com/zh-cn/documentation/sles10/book_sle_reference/data/sec.kerbadmin.ldap.html)
- kerberos与openldap整合 (http://wenku.baidu.com/view/fe7c82757fd5360cba1adbe7.html)
- LDAP配置示例 (http://ovirt-china.org/mediawiki/index.php/LDAP%E9%85%8D%E7%BD%AE%E7%A4%BA%E4%BE%8B)
- centos下yum安装配置openldap 2.4.23-32外送svn的apache下配置 (http://kinggoo.com/openldapinstallconf.htm)
- Integrating LDAP and Kerberos: Part Two (LDAP) (http://www.linux-mag.com/id/4765/)
- Debian GNU and Ubuntu: Setting up MIT Kerberos (http://techpubs.spinlocksolutions.com/dklar/kerberos.html)