一、Kerberos安装

1、kerberos 服务器端

hpt2作为主节点:

```
yum install krb5-server krb5-libs krb5-auth-dialog krb5-workstation
```

修改/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 = master
  dns_lookup_realm = false
  dns_lookup_kdc = false
  ticket_lifetime = 24h
  renew_lifetime = 7d
  forwardable = true
  renewable = true

[realms]
  master = {
    kdc = hpt2
    admin_server = hpt2
  }

[domain_realm]
  hpt1 = master
  hpt2 = master
  hpt3 = master
```

修改/var/kerberos/krb5kdc/kdc.conf

```
[kdcdefaults]
```

```
  kdc_ports = 88
  kdc_tcp_ports = 88

[realms]
  master = {
    #master_key_type = aes256-cts
    acl_file = /var/kerberos/krb5kdc/kadm5.acl
    dict_file = /usr/share/dict/words
    max_renewable_life = 7d
    max_life = 1d
    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
  }
```

修改/var/kerberos/krb5kdc/kadm5.acl

```
*/admin@master   *
```

以上三个文件配置完毕后，只需拷贝krb5.conf到集群中其他机器上即可。

```
scp krb5.conf hpt1:/etc/
scp krb5.conf hpt3:/etc/
```

2、kerberos客户端

hpt1、hpt2作为客户端

```
yum install krb5-workstation krb5-libs krb5-auth-dialog
```

3、关于AES-256加密

oracle官网下载jce_policy-8.zip，解压，将local_policy.jar、US_export_policy.jar拷贝到$JAVA_HOME/jre/lib/security目录下

4、创建数据库

在 hpt2 上运行初始化数据库命令。其中 -r 指定对应 realm，初始密码123456

$ kdb5_util create -r master -s

出现 Loading random data 的时候另开个终端执行点消耗CPU的命令如 cat /dev/sda > /dev/urandom 可以加快随机数采集。该命令会在 /var/kerberos/krb5kdc/ 目录下创建 principal 数据库。

如果遇到数据库已经存在的提示，可以把 /var/kerberos/krb5kdc/ 目录下的 principal 的相关文件都删除掉。默认的数据库名字都是 principal。可以使用 -d 指定数据库名字。

5、启动服务

在hpt2上运行如下命令：

$ chkconfig --level 35 krb5kdc on $ chkconfig --level 35 kadmin on $ service krb5kdc start $ service kadmin start

6、创建 kerberos 管理员

关于 kerberos 的管理，可以使用 kadmin.local 或 kadmin，至于使用哪个，取决于账户和访问权限：

如果有访问 kdc 服务器的 root 权限，但是没有 kerberos admin 账户，使用 kadmin.local

如果没有访问 kdc 服务器的 root 权限，但是用 kerberos admin 账户，使用 kadmin

在 hpt2 上创建远程管理的管理员：

在KDC server主机上，创建一个名为『hadoop』的principal，并将其密码设为『hadoop』。

执行命令：

[root@hpt2 /]# kadmin.local

Authenticating as principal root/admin@master with password.

kadmin.local: addprinc -pw hadoop hadoop/admin@master

通过执行kadmin.local中的listprincs命令可以看到创建了一个名为『hadoop/admin@master』的principal：

kadmin.local: listprincs

K/M@master

hadoop/admin@master

kadmin/admin@master

kadmin/changepw@master

kadmin/hpt2@master

krbtgt/master@master

principal的名字的第二部分是admin,那么该principal就拥有administrative privileges

这个账号将会被CDH用来生成其他用户/服务的principal

登录到管理员账户：如果在本机上，可以通过kadmin.local直接登录。其它机器的，先使用kinit进行验证。

```
[root@hpt2 app]# kadmin.local
Authenticating as principal root/admin@master with password.
kadmin.local:
```

```
[root@hpt3 upgrade]# kinit hadoop/admin
Password for hadoop/admin@master:
[root@hpt3 upgrade]# kadmin
Authenticating as principal hadoop/admin@master with password.
Password for hadoop/admin@master:
kadmin:
```

7、测试 kerberos

查看principals $ kadmin: list_principals

添加一个新的 principal kadmin: addprinc user1 WARNING: no policy specified for user1@JAVACHEN.COM; defaulting to no policy Enter password for principal "user1@JAVACHEN.COM": Re-enter password for principal "user1@JAVACHEN.COM": Principal "user1@JAVACHEN.COM" created.

删除 principal kadmin: delprinc user1 Are you sure you want to delete the principal "user1@JAVACHEN.COM"? (yes/no): yes Principal "user1@JAVACHEN.COM" deleted. Make sure that you have removed this principal from all ACLs before reusing. kadmin: exit

列出Kerberos中的所有认证用户，即principals kadmin.local -q "list_principals" # 添加认证用户，需要输入密码 kadmin.local -q "addprinc user1" # 使用该用户登录，获取身份认证，需要输入密码 kinit user1 # 查看当前用户的认证信息ticket klist # 更新ticket kinit -R # 销毁当前的ticket kdestroy # 删除认证用户 kadmin.local -q "delprinc user1"

8、CDH启用Kerberos

在CM的界面上点击启用Kerberos，启用的时候需要确认几个事情：

1. KDC已经安装好并且正在运行

2. 将KDC配置为允许renewable tickets with non-zerolifetime

- 在之前修改kdc.conf文件的时候已经添加了kdc_tcp_ports、max_life和max_renewable_life这个三个选项

3. 在Cloudera Manager Server上安装openldap-clients

4. 为Cloudera Manager创建一个principal，使其能够有权限在KDC中创建其他的principals，就是上面创建的Kerberos管理员账号

确定完了之后点击continue，进入下一页进行配置，要注意的是：这里的『Kerberos Encryption Types』必须跟KDC实际支持的加密类型匹配（即kdc.conf中的值）

这里使用了默认的aes256-cts

注意，这里的『Kerberos Encryption Types』必须和/etc/krb5.conf中的default_tgs_enctypes、default_tkt_enctypes和permitted_enctypes三个选项的值对应起来，不然会出现集群服务无法认证通过的情况！

异常信息：javax.security.auth.login.LoginException: No supported encryption types listed in default_tkt_enctypes

点击continue，进入下一页，这一页中可以不勾选『Manage krb5.conf through Cloudera Manager』

注意，如果勾选了这个选项就可以通过CM的管理界面来部署krb5.conf，**但是实际操作过程中发现有些配置仍然需要手动修改该文件并同步**

点击continue，进入下一页，输入Cloudera Manager Principal的管理员账号和密码，**注意输入账号的时候要使用@前要使用全称，xiaohei/admin**

点击continue，进入下一页，导入KDC Account Manager Credentials

点击continue，进入下一页，restart cluster并且enable Kerberos

之后CM会自动重启集群服务，启动之后会会提示Kerberos已启用

这个过程中CM会自动在Kerberos的数据库中创建各个节点中各个账户对应的principle

可以使用 kadmin.local -q "list_principals"查看，，格式为username/hostname@XIAOHEI.INFO，例如hdfs/hadoop-10-0-8-124@XIAOHEI.INFO

在CM上启用Kerberos的过程中，CM会自动做以下的事情：

1. 集群中有多少个节点，每个账户都会生成对应个数的principal
2. 为每个对应的principal创建keytab
3. 部署keytab文件到指定的节点中
4. 在每个服务的配置文件中加入有关Kerberos的配置

其中包括Zookeeper服务所需要的jaas.conf和keytab文件都会自动设定并读取，如果用户仍然手动修改了Zookeeper的服务，要确保这两个文件的路径和内容正确性

keytab是包含principals和加密principal key的文件

keytab文件对于每个host是唯一的，因为key中包含hostname

keytab文件用于不需要人工交互和保存纯文本密码，实现到kerberos上验证一个主机上的principal

启用之后访问集群的所有资源都需要使用相应的账号来访问，否则会无法通过Kerberos的authenticatin

9、创建HDFS超级用户

此时直接用CM生成的principal访问HDFS会失败，因为那些自动生成的principal的密码是随机的，用户并不知道，而通过命令行的方式访问HDFS需要先使用kinit来登录并获得ticket

所以使用kinit hdfs/hptl@master需要输入密码的时候无法继续

用户可以通过创建一个hdfs@master的principal并记住密码从命令行中访问HDFS

```
# 需要输入两遍密码 kadmin.local -q "addprinc hdfs"
```

先使用

```
kinit hdfs@master
```

登录之后就可以通过认证并访问HDFS

默认hdfs用户是超级用户

10、为每个用户创建principal

当集群运行Kerberos后，每一个Hadoop user都必须有一个principal或者keytab来获取Kerberos credentials

（即使用密码的方式或者使用keytab验证的方式）

这样才能访问集群并使用Hadoop的服务

也就是说，如果Hadoop集群存在一个名为hdfs@XIAOHEI.INFO的principal

那么在集群的每一个节点上应该存在一个名为hdfs的Linux用户

同时，在HDFS中的目录/user要存在相应的用户目录（即/user/hdfs），且该目录的owner和group都要是hdfs

至此，集群上的服务都启用了Kerberos的安全认证

11、确认HDFS可以正常使用

登录到某一个节点后，切换到hdfs用户，然后用kinit来获取credentials

现在用’hadoop dfs -ls /’应该能正常输出结果

用kdestroy销毁credentials后，再使用hadoop dfs -ls /会发现报错

12、确认可以正常提交MapReduce job

获取了hdfs的证书后，提交一个PI程序，如果能正常提交并成功运行，则说明Kerberized Hadoop cluster在正常工作

```
hadoop jar /opt/cloudera/parcels/CDH-5.7.0-1.cdh5.7.0.p0.45/jars/hadoop-examples.jar pi 10 1000
```

13、生成keytab

在KDC节点，生成独立keytab

```
kadmin.local -q "xst  -k hpt.keytab hpt "
```

创建包含多个用户的keytab

```
kadmin.local -q "xst  -k hpt.keytab hpt hdfs"
```

显示keytab的文件列表

```
klist -ket hpt.keytab
```

注册

```
kinit -kt hpt.keytab hpt
```

14、beeline连接hive、impala

hive： beeline -u 'jdbc:hive2://hpt3:10000/default;principal=hive/hpt3@master'

impala：

遇到的问题

1、 hdfs用户被禁止运行 **YARN container**

16/10/28 14:42:40 INFO mapreduce.Job: Job job_1477622828980_0003 failed with state FAILED due to: Application application_1477622828980_0003 failed 2 times due to AM Container for appattempt_1477622828980_0003_000002 exited with
  exitCode: -1000
For more detailed output, check application tracking page:http://hpt1:8088/proxy/application_1477622828980_0003/Then, click on links to logs of each attempt.
Diagnostics: Application application_1477622828980_0003 initialization failed (exitCode=255) with output: main : command provided 0
main : run as user is hdfs
main : requested yarn user is hdfs
Requested user hdfs is banned

原因：

yarn的设置中将hdfs用户禁用了

解决方法：

修改Clouder关于这个该项的设置，Yarn->配置->banned.users 将hdfs用户移除

2、hue无法运行sqoop脚本

32446 [uber-SubtaskRunner] INFO org.apache.sqoop.hive.HiveImport - Loading uploaded data into Hive Heart beat Intercepting System.exit(1) <<< Invocation of Main class completed <<< Failing Oozie Launcher, Main class [org.apache.oozie.action.hadoop.SqoopMain], exit code [1] Oozie Launcher failed, finishing Hadoop job gracefully Oozie Launcher, uploading action data to HDFS sequence file: hdfs://ns1/user/yarn/oozie-oozi/0000048-161102180608221-oozie-oozi-W/sqoop-9e83--sqoop/action-data.seq Oozie Launcher ends

解决方法：

在workflow sqoop组件中的凭据中勾选hcat

二、LDAP安装

1、 安装openldap

在hpt2上安装

$ yum install -y db4 db4-utils db4-devel cyrus-sasl* krb5-server-ldap

$ yum install -y openldap openldap-servers openldap-clients openldap-devel

有两个文件要复制：slapd的配置文件和数据库文件，将openldap-servers自带的example复制到相应目录：

cp /usr/share/openldap-servers/slapd.conf.obsolete /etc/openldap/slapd.conf

cp /usr/share/openldap-servers/DB_CONFIG.example /var/lib/ldap/DB_CONFIG

2、LDAP 服务端配置

使用slappasswd创建LDAP管理员密码，这个命令不会直接将密码写入配置，运行slappasswd后输入两次密码，会返回一串密文，复制下这个密文。

root@vm ~]# slappasswd

New password:

Re-enter new password:

{SSHA}wbUwoQkhy5871pE7KVvfkGW0KJFr3O/T

编辑 vi /etc/openldap/slapd.conf，找到"database bdb"，按照自己的需求更改下面的：

suffix "dc=my-domain,dc=com" rootdn "cn=admin,dc=my-domain,dc=com" //管理员为

admin rootpw {SSHA}wbUwoQkhy5871pE7KVvfkGW0KJFr3O/T //复制的管理员的密码，也支持明文

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/

在/etc/openldap/slapd.conf加入

include /etc/openldap/schema/kerberos.schema

chown -R ldap:ldap /etc/openldap chown -R ldap:ldap /var/lib/ldap

配置双主Mirror Mode模式（可选操作）

以上1、2操作在hpt1上同样执行

修改hpt1配置：

modulepath /usr/lib/openldap

modulepath /usr/lib64/openldap

moduleload syncprov.la

```
# syncrepl directives
overlay syncprov
syncprov-checkpoint 100 10
syncprov-sessionlog 100
serverID       1
syncrepl       rid=002
                        provider=ldap://hpt2
                        bindmethod=simple
                        binddn="cn=admin,dc=my-domain,dc=com"
                        credentials=baofoo#64
                        searchbase="dc=my-domain,dc=com"
                        schemachecking=on
                        type=refreshAndPersist
                        retry="60 +"
mirrormode on
```

修改hpt2配置：

```
modulepath /usr/lib/openldap
modulepath /usr/lib64/openldap
moduleload syncprov.la

# syncrepl directives
overlay syncprov
syncprov-checkpoint 100 10
syncprov-sessionlog 100
serverID       2
syncrepl       rid=002
                        provider=ldap://hpt1
                        bindmethod=simple
                        binddn="cn=admin,dc=my-domain,dc=com"
                        credentials=baofoo#64
                        searchbase="dc=my-domain,dc=com"
                        schemachecking=on
                        type=refreshAndPersist
                        retry="60 +"
```

mirrormode on

3、测试并生成配置文件

rm -rf /etc/openldap/slapd.d/* //删除原文件 service slapd start //生成bdb文件 slaptest -f /etc/openldap/slapd.conf -F /etc/openldap/slapd.d //生成配置文件 chown -R ldap:ldap /etc/openldap/slapd.d

4、配置完成重启服务

service slapd restart chkconfig slapd on //设置开机自启动

经过上面的配置后，openldap server就配置好了。

查看状态，验证服务端口：

```
[root@hpt2 openldap]# ps aux | grep slapd | grep -v grep
ldap         32724   0.1   0.3 584120 50360 ?              Ssl   13:54     0:00
/usr/sbin/slapd -h   ldap:/// ldapi:/// -u ldap
[root@hpt2 openldap]# netstat -tunlp   | grep :389
tcp               0          0 0.0.0.0:389
                                0.0.0.0:*
                                    LISTEN          32724/slapd
```

查看LDAP数据库结构：

ldapsearch -x -H ldap://127.0.0.1 -b 'dc=my-domain,dc=com'

会返回类似：

```
# extended LDIF
#
# LDAPv3
# base <dc=my-domain,dc=com> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
# search result
search: 2
result: 32 No such object
```

```
# numResponses: 1
```

5、LDAP 和 Kerberos

为了使Kerberos能够绑定到OpenLDAP服务器，需要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（一般为ldap）：

```
kadmin.local -q "addprinc ldapadmin@master" kadmin.local -q "addprinc -randkey ldap/hpt2@master" kadmin.local -q "ktadd -k /etc/openldap/ldap.keytab ldap/hpt2@master" chown ldap:ldap /etc/openldap/ldap.keytab && chmod 640 /etc/openldap/ldap.keytab
```

使用ldapadmin用户测试：

```
kinit ldapadmin
```

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
```

重启 service slapd restart

6、配置并迁移系统用户

配置好的LDAP数据库是空的，需要将系统上的用户导入到LDAP数据库中。需要用migrationtools将系统用户转换为LDAP能识别的ldif文件。

先新建一个test用户作为测试用户：

```
useradd test echo "pwdpwd" |passwd --stdin test
```

安装migrationtools：

```
yum install migrationtools
```

配置migrationtools：

编辑/usr/share/migrationtools/migrate_common.ph ，按需更改下面两行：

```
$DEFAULT_MAIL_DOMAIN = "my-domain.com"; $DEFAULT_BASE = "dc=my-domain,dc=com";
```

生成模板文件：

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

生成ldif文件： /usr/share/migrationtools/migrate_passwd.pl /etc/passwd >/opt/passwd.ldif

/usr/share/migrationtools/migrate_group.pl /etc/group >/opt/group.ldif

【ldapadd】添加节点:

-x 表示密码验证;

-D <DN> 指定管理员的DN;

-w <PASSWORD> 指定管理员的密碼;

-W 提示输入管理员的密碼;

-f <FILE> 指定LDIF文件路径;

-H <URL> 指定LDAP服务器的URL(例如本机为 ldap://localhost/);

LDAP的主要名词解释

dc：（Domain Component）域

ObjectClass：对象类（不同的对象类存在某些不同的属性，根据自己需要选择对象类）

dn：(Distinguished Name)节点绝对路径,例如:uid=admin,ou=users,dc=eruipan,dc=com

o:(Organizational)组织

ou：(Organizational Unit)组织单位

cn：(Common Name)通用名（继承自person对象的对象类必须有值的属性，否则无法创建）

sn：（surname）全名（继承自person对象的对象类必须有值的属性，否则无法创建）

其他需要用到的属性

mail：电子邮件

userpassword：用户密码

uid：唯一标识（如果使用uid验证）

如果有需要，也可以编辑passwd.ldif和group.ldif去掉不需要的条目。

将生成的ldif导入到LDAP数据库：

ldapadd -x -D "cn=admin,dc=my-domain,dc=com" -W -f /opt/base.ldif ldapadd -x -D "cn=admin,dc=my-domain,dc=com" -W -f /opt/passwd.ldif ldapadd -x -D "cn=admin,dc=my-domain,dc=com" -W -f /opt/group.ldif

查询新添加的 test 用户：

ldapsearch -LLL -x -D 'cn=admin,dc=my-domain,dc=com' -W -b 'dc=my-domain,dc=com' 'cn=develop'

修改用户密码：

```
ldappasswd -x -D 'cn=admin,dc=my-domain,dc=com' -w admin
"uid=hdfs,ou=people,dc=my-domain,dc=com" -S
```

```
[root@hpt2 opt]# ldapsearch -LLL -x -D 'cn=admin,dc=my-domain,dc=com' -w admin -
b 'dc=my-domain,dc=com' 'cn=test'
dn: uid=test,ou=People,dc=my-domain,dc=com
uid: test
cn: test
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword:: e2NyeXB0fSQ2JHdVUWxjTFFUJHZnVHo4VOx6bmO5eDNIeW11WE5kUWRISTVMSHp
  qeWxLT1pVOVZoeUZzcm5ON1hSQVlQdDZZUkliW1dCWUh6YU1YNXV3Q3BWRmdObW5KSUo3OEdNaUOv
shadowLastChange: 17113
shadowMin: 0
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 1002
gidNumber: 1002
homeDirectory: /home/test

dn: cn=test,ou=Group,dc=my-domain,dc=com
objectClass: posixGroup
objectClass: top
cn: test
userPassword:: e2NyeXB0fXg=
gidNumber: 1002
```

7、LDAP客户端配置

在其他节点上运行

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

修改 /etc/openldap/ldap.conf 以下两个配置

BASE dc=my-domain,dc=com URI ldap://hpt2

然后，运行下面命令测试：

#先删除 ticket $ kdestroy

[root@hpt2 openldap]# ldapsearch -b 'dc=my-domain,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 (Credentials cache file '/tmp/krb5cc_0' not found)

重新获取 ticket：

$ kinit ldapadmin $ ldapsearch -b 'dc=my-domain,dc=com' # 没有报错了 [root@hpt2 openldap]# ldapwhoami

SASL/GSSAPI authentication started

SASL username: ldapadmin@master

SASL SSF: 56

SASL data security layer installed.

dn:uid=ldapadmin,cn=gssapi,cn=auth

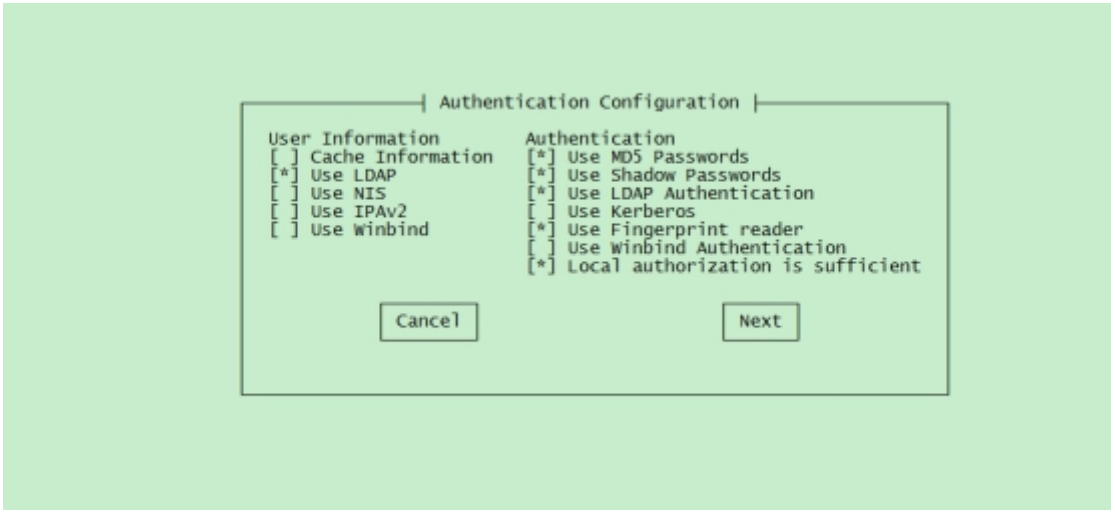
配置LDAP客户机，统一管理Linux用户（可选操作）

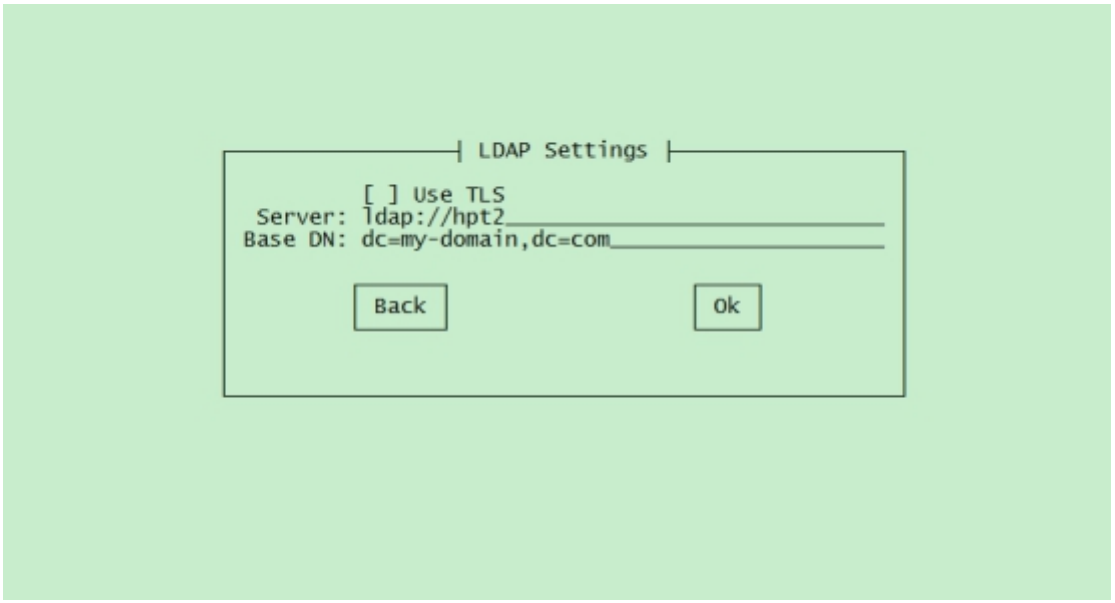
1.安装ldap client认证需要的pam软件包

yum install nss-pam-ldapd pam_ldap sssd -y

2.配置客户端账户认证方式

[root@hpt3 ~]# authconfig-tui





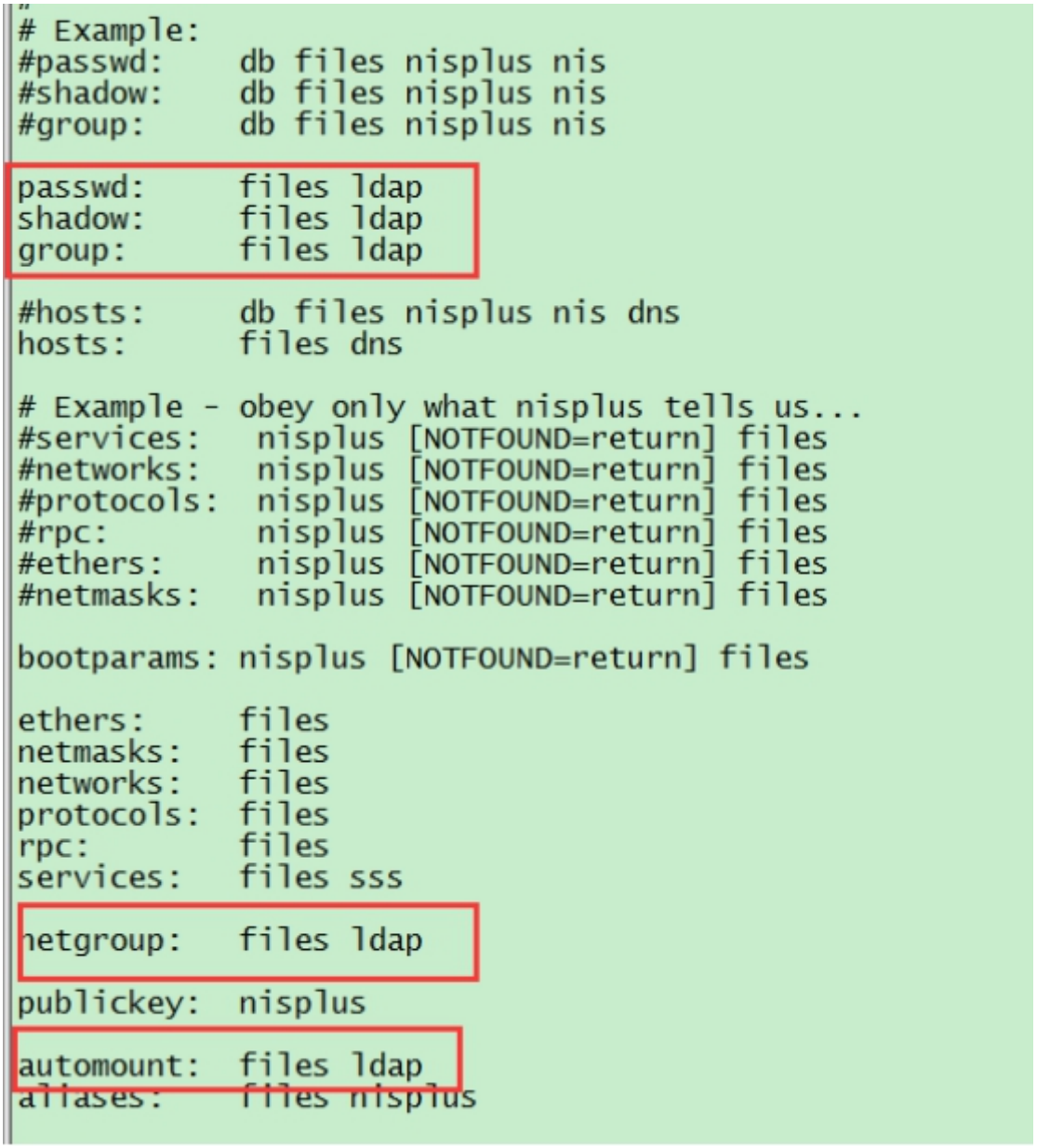
3. 修改sssd配置文件并重启服务

vi /etc/sssd/sssd.conf

添加enumerate=true (因该不是必须的，但是加上为妙)

service sssd restart 重启服务

4. 修改 vi /etc/nsswitch.conf



5.编辑系统认证文件，保证采用ldap来认证

编辑/etc/pam.d/system-auth和/etc/pam.d/password-auth文件，将pam_sss.so更改为pam_ldap.so，用于实现用户su之间切换

分别编辑：　　vi /etc/pam.d/system-auth

　　vi /etc/pam.d/password-auth

```
session     required      pam_unix.so
session     optional      pam_sss.so
[root@hpt3 sssd]# vim /etc/pam.d/{system,password}-auth
2 files to edit
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      pam_env.so
auth        sufficient    pam_fprintd.so
auth        sufficient    pam_unix.so nullok try_first_pass
auth        requisite     pam_succeed_if.so uid >= 500 quiet
auth        sufficient    pam_ldap.so use_first_pass
auth        required      pam_deny.so

account     required      pam_unix.so broken_shadow
account     sufficient    pam_localuser.so
account     sufficient    pam_succeed_if.so uid < 500 quiet
account     [default=bad success=ok user_unknown=ignore] pam_ldap.so
account     required      pam_permit.so

password    requisite     pam_cracklib.so try_first_pass retry=3 type=
password    sufficient    pam_unix.so md5 shadow nullok try_first_pass use_authtok
password    sufficient    pam_ldap.so use_authtok
password    required      pam_deny.so

session     optional      pam_keyinit.so revoke
session     required      pam_limits.so
session     [success=1 default=ignore] pam_succeed_if.so service in crond quiet use_uid
session     required      pam_unix.so
session     optional      pam_ldap.so
~
~
```

重启服务 service nslcd restart

8、配置 Hive 集成 LDAP

在hive-site.xml中加入以下配置：

```
<property>
    <name>hive.server2.authentication</name>
    <value>LDAP</value>
</property>
<property>
    <name>hive.server2.authentication.ldap.url</name>
    <value>ldap://hpt2</value>
</property>
<property>
    <name>hive.server2.authentication.ldap.baseDN</name>
    <value>ou=people,dc=my-domain,dc=com</value>
</property>
```

重启Hive和Yarn服务,进入beeline测试：

```
[root@hpt3 conf]# beeline
beeline> !connect jdbc:hive2://hpt3:10000/default
scan complete in 4ms
Connecting to jdbc:hive2://hpt3:10000/default
```

```
Enter username for jdbc:hive2://hpt3:10000/default: hdfs
Enter password for jdbc:hive2://hpt3:10000/default: ****
Connected to: Apache Hive (version 1.1.0-cdh5.7.0)
Driver: Hive JDBC (version 1.1.0-cdh5.7.0)
Transaction isolation: TRANSACTION_REPEATABLE_READ
0: jdbc:hive2://hpt3:10000/default>
```

9、配置 Impala 集成 LDAP

Impala中可以同时使用Kerberos+LDAP的认证方式，所以在已经启用Kerberos的情况下启用LDAP可以正常工作

在Impala配置页中：

启用 LDAP 身份验证选项设置为true

启用 LDAP TLS 选项设置为true

Impala 命令行参数高级配置代码段（安全阀）中添加-ldap_baseDN=ou=people,dc=my-domain,dc=com

重启Impala服务

使用impala-shell测试LDAP账号：

```
[root@hpt3 conf]# impala-shell -l -u hpt --auth_creds_ok_in_clear
Starting Impala Shell using LDAP-based authentication
LDAP password for hpt:
```

使用beeline测试LDAP账号：

```
beeline -u "jdbc:hive2://hpt3:21050/default;" -n hpt -p hpt
```

10、Hue集成LDAP

在Hue中配置LDAP可以让Hue直接使用LDAP所管理的账号而不必在Hue中重新管理

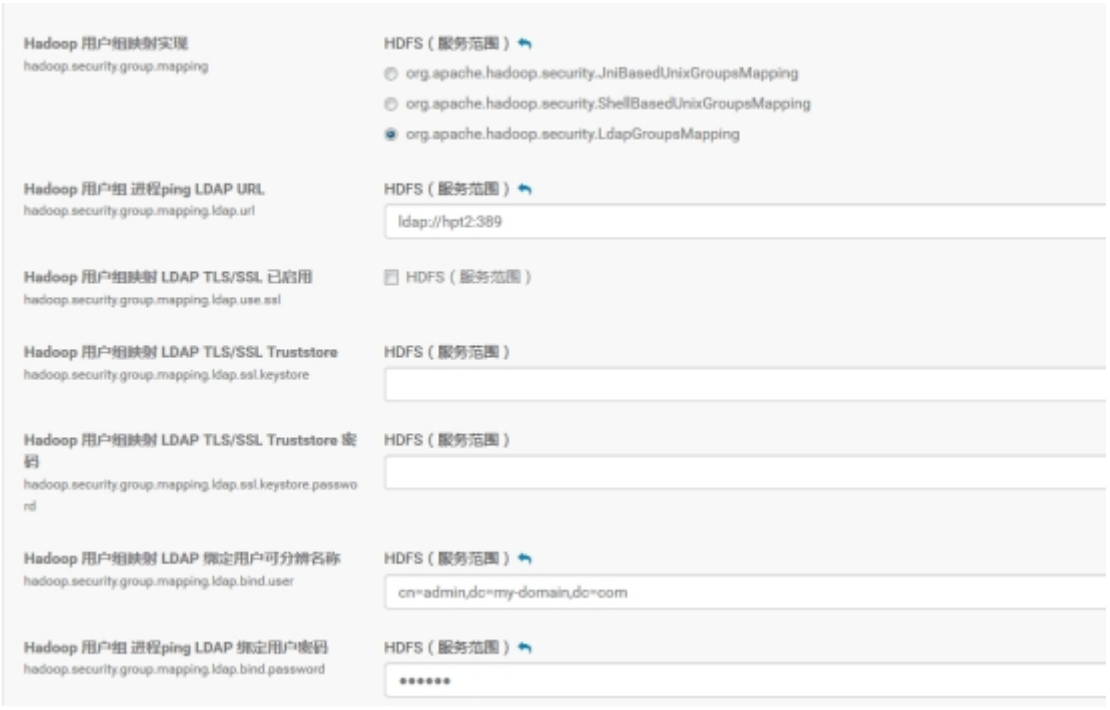
在Hue的配置页面中修改

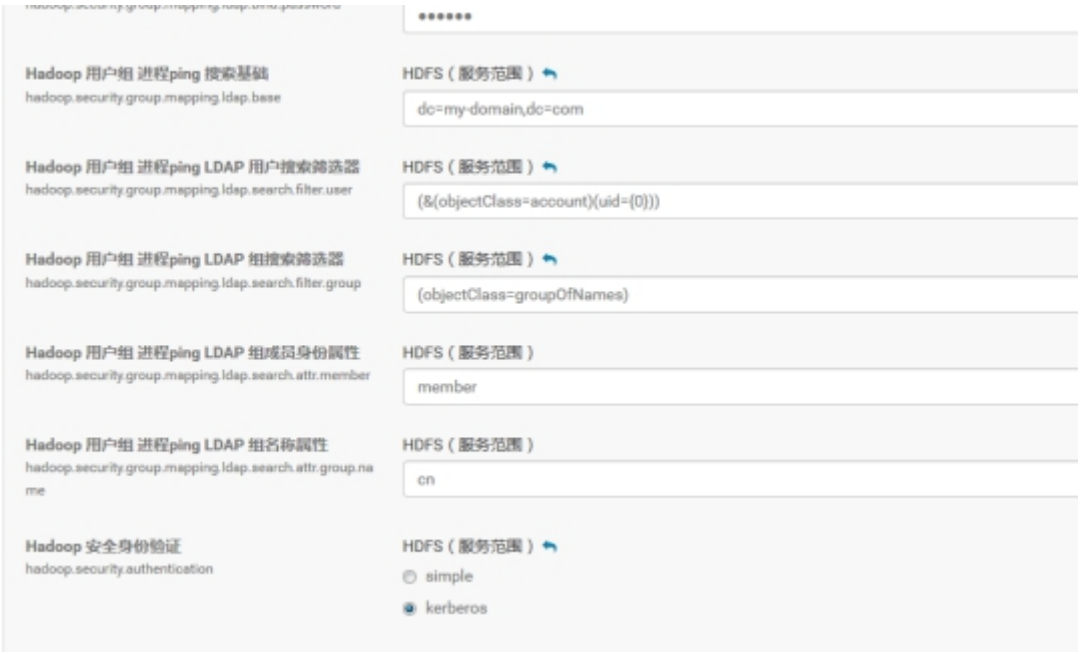
身份验证后端/backend为desktop.auth.backend.LdapBackend

登录时创建 LDAP 用户/create_users_on_login 设置为True

使用搜索绑定身份验证/search_bind_authentication 设置为False

配置hadoop ldap映射，测试使用，生产未配置


Hadoop 用户组映射实现
hadoop.security.group.mapping
HDFS（服务范围）
org.apache.hadoop.security.JniBasedUnixGroupsMapping
org.apache.hadoop.security.ShellBasedUnixGroupsMapping
org.apache.hadoop.security.LdapGroupsMapping
Hadoop 用户组 进程ping LDAP URL
hadoop.security.group.mapping.ldap.url
HDFS（服务范围）
ldap://hpt2:389
Hadoop 用户组映射 LDAP TLS/SSL 已启用
hadoop.security.group.mapping.ldap.use.ssl
HDFS（服务范围）
Hadoop 用户组映射 LDAP TLS/SSL Truststore
hadoop.security.group.mapping.ldap.ssl.keystore
HDFS（服务范围）
Hadoop 用户组映射 LDAP TLS/SSL Truststore 密码
hadoop.security.group.mapping.ldap.ssl.keystore.password
HDFS（服务范围）
Hadoop 用户组映射 LDAP 绑定用户可分辨名称
hadoop.security.group.mapping.ldap.bind.user
HDFS（服务范围）
cn=admin,dc=my-domain,dc=com
Hadoop 用户组 进程ping LDAP 绑定用户密码
hadoop.security.group.mapping.ldap.bind.password
HDFS（服务范围）
••••••


••••••
Hadoop 用户组 进程ping 搜索基础
hadoop.security.group.mapping.ldap.base
HDFS（服务范围）
dc=my-domain,dc=com
Hadoop 用户组 进程ping LDAP 用户搜索筛选器
hadoop.security.group.mapping.ldap.search.filter.user
HDFS（服务范围）
(&(objectClass=account)(uid={0}))
Hadoop 用户组 进程ping LDAP 组搜索筛选器
hadoop.security.group.mapping.ldap.search.filter.group
HDFS（服务范围）
(objectClass=groupOfNames)
Hadoop 用户组 进程ping LDAP 组成员身份属性
hadoop.security.group.mapping.ldap.search.attr.member
HDFS（服务范围）
member
Hadoop 用户组 进程ping LDAP 组名称属性
hadoop.security.group.mapping.ldap.search.attr.group.name
HDFS（服务范围）
cn
Hadoop 安全身份验证
hadoop.security.authentication
HDFS（服务范围）
simple
kerberos