OpenLDAP

Présentation :

Lightweight Directory Access Protocol (LDAP) est à l'origine un protocole permettant l'interrogation et la modification des services d'annuaire. Ce protocole repose sur TCP/IP. Il a cependant évolué pour représenter une norme pour les systèmes d'annuaires, incluant un modèle de données, un modèle de nommage, un modèle fonctionnel basé sur le protocole LDAP, un modèle de sécurité et un modèle de réplication. C'est une structure arborescente dont chacun des nœuds est constitué d'attributs associés à leurs valeurs. LDAP est moins complexe que le modèle X.500 édicté par l'UIT-T.

L'objectif de ce TP est d'installer et configurer OpenLDAP avec un client graphique pour la gestion de l'annuaire mais aussi de s'authentifier sur un client Linux avec authentification LDAP.

Pré requis :

- ➔ Ordinateurs sur Linux (ici, on utilisera la Debian 8.3).
- ➔ Avoir une connexion internet.
- ➔ On utilisera l'utilisateur root
- ➔ L'adresse IP du serveur est 192.168.1.134/24 et le client est 192.168.1.135/24

Sommaire :

I. Installation et configuration du serveur

Nous allons installer les paquets nécessaires :

```
root@debian8:~# apt-get install slapd ldap-utils
```

Nous allons configurer ldap maintenant et remplir ce fichier :

```
root@debian8:~# nano /etc/ldap/ldap.conf
```

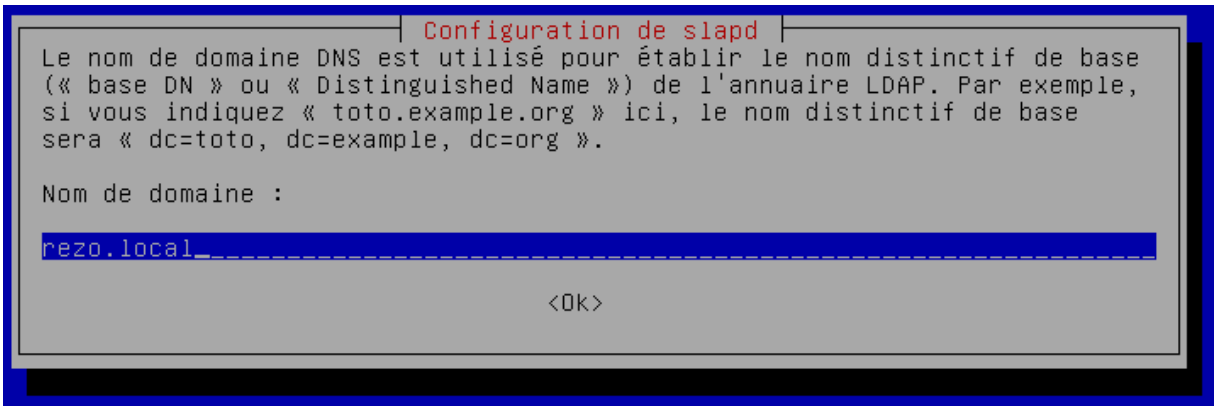
Puis par ceci :

```
  GNU nano 2.2.6          Fichier : /etc/ldap/ldap.conf                Modifié

#
# LDAP Defaults
#

# See ldap.conf(5) for details
# This file should be world readable but not world writable.

BASE    dc=rezo,dc=local
URI     ldap://192.168.1.134

#SIZELIMIT      12
#TIMELIMIT      15
#DEREF          never
```

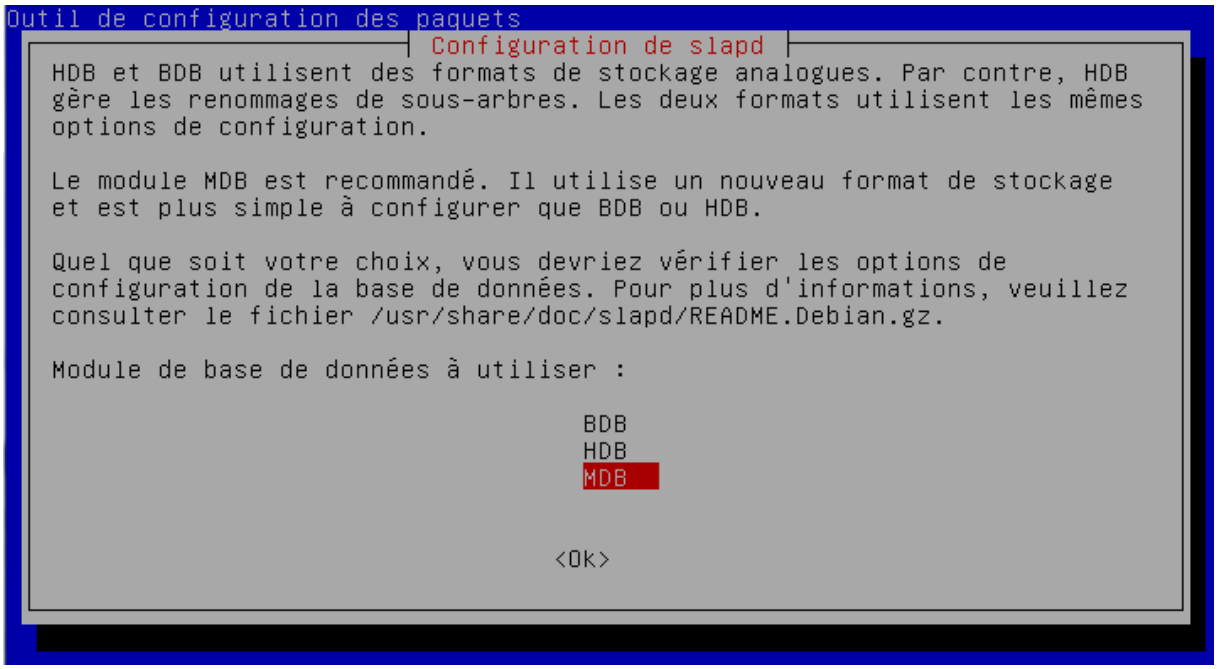
On va maintenant reconfigurer OpenLDAP :

```
root@debian8:~# dpkg-reconfigure slapd
```

On sélectionne « Non » puis on rentre le nom de domaine :



Après avoir rentré le même mot de passe lors de l'installation des paquets, nous allons choisir MDB.



Enfin, on ne garde pas l'ancienne base mais on la déplace. De plus, nous n'allons pas utiliser la version 2 du protocole LDAP.

On peut ainsi tester le serveur LDAP avec cette commande :

```
root@debian8:~# ldapsearch -x
```

II. Administration du serveur

Nous allons installer un serveur web avec php pour que phpLDAPadmin puisse fonctionner :

```
root@debian8:~# apt-get install apache2 php5 phpldapadmin
```

Pour des raisons de sécurité, on va modifier les droits d'accès :

```
root@debian8:~# chown -R www-data:www-data /etc/phpldapadmin/
root@debian8:~# chmod 640 /etc/phpldapadmin/config.php
root@debian8:~# chown -R www-data:www-data /usr/share/phpldapadmin/
```

Nous allons configurer un fichier php pour que phpLDAPAdmin puissent fonctionner sans problème :

```
root@debian8:~# nano /etc/phpldapadmin/config.php
```

Première modification :

```
/* A convenient name that will appear in the tree viewer and throughout
   phpLDAPadmin to identify this LDAP server to users. */
$servers->setValue('server','name','192.168.1.134');
```

Seconde modification :

```
/* Array of base DNs of your LDAP server. Leave this blank to have phpLDAPadmin
   auto-detect it for you. */
$servers->setValue('server','base',array('dc=rezo,dc=local'));
```

Dernière modification :

```
/* The DN of the user for phpLDAPadmin to bind with. For anonymous binds or
   'cookie','session' or 'sasl' auth_types, LEAVE THE LOGIN_DN AND LOGIN_PASS
   BLANK. If you specify a login_attr in conjunction with a cookie or session
   auth_type, then you can also specify the bind_id/bind_pass here for searching
   the directory for users (ie, if your LDAP server does not allow anonymous
   binds. */
$servers->setValue('login','bind_id','cn=admin,dc=rezo,dc=local');
#  $servers->setValue('login','bind_id','cn=Manager,dc=example,dc=com');
```

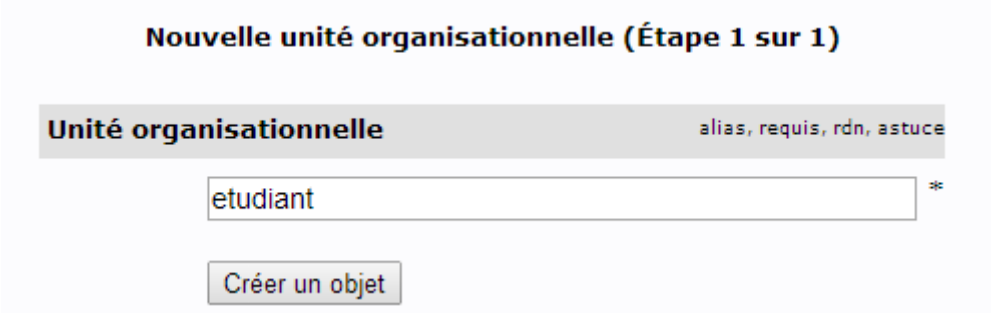
Enfin, on modifie la timezone :

```
/* Our local timezone
   This is to make sure that when we ask the system for the current time, we
   get the right local time. If this is not set, all time() calculations will
   assume UTC if you have not set PHP date.timezone. */
// $config->custom->appearance['timezone'] = null;
#  $config->custom->appearance['timezone'] = 'Europe/Paris';
```

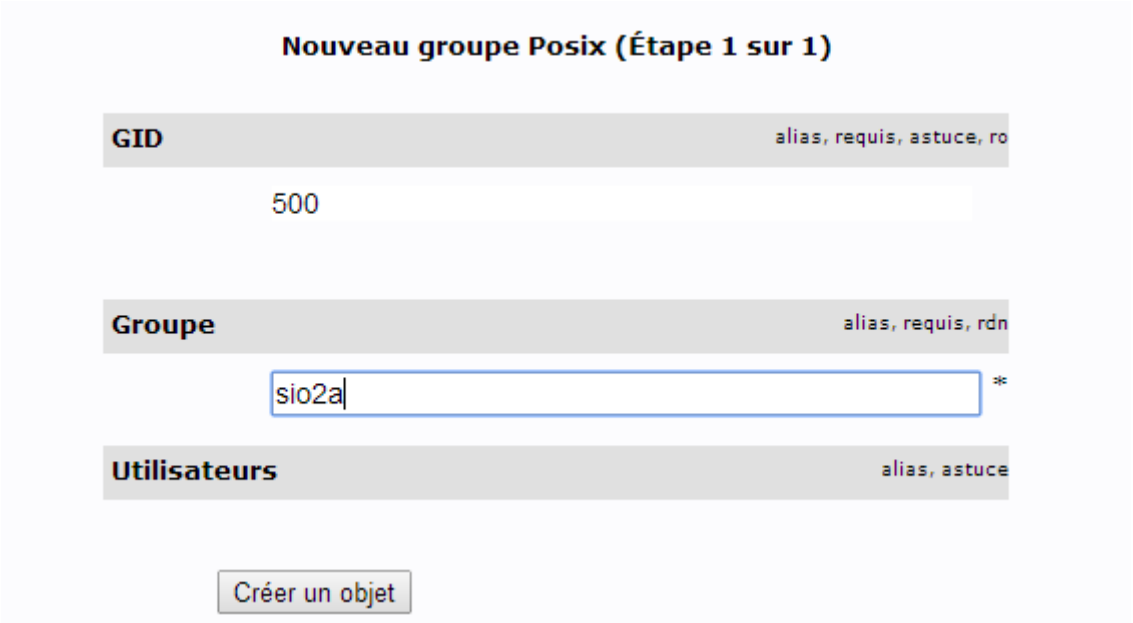
On relance le service apache2 et dans un navigateur on ouvre cette url :

http://@IP_du_serveur/phpldapadmin (on se connecte avec le mot de passe root)

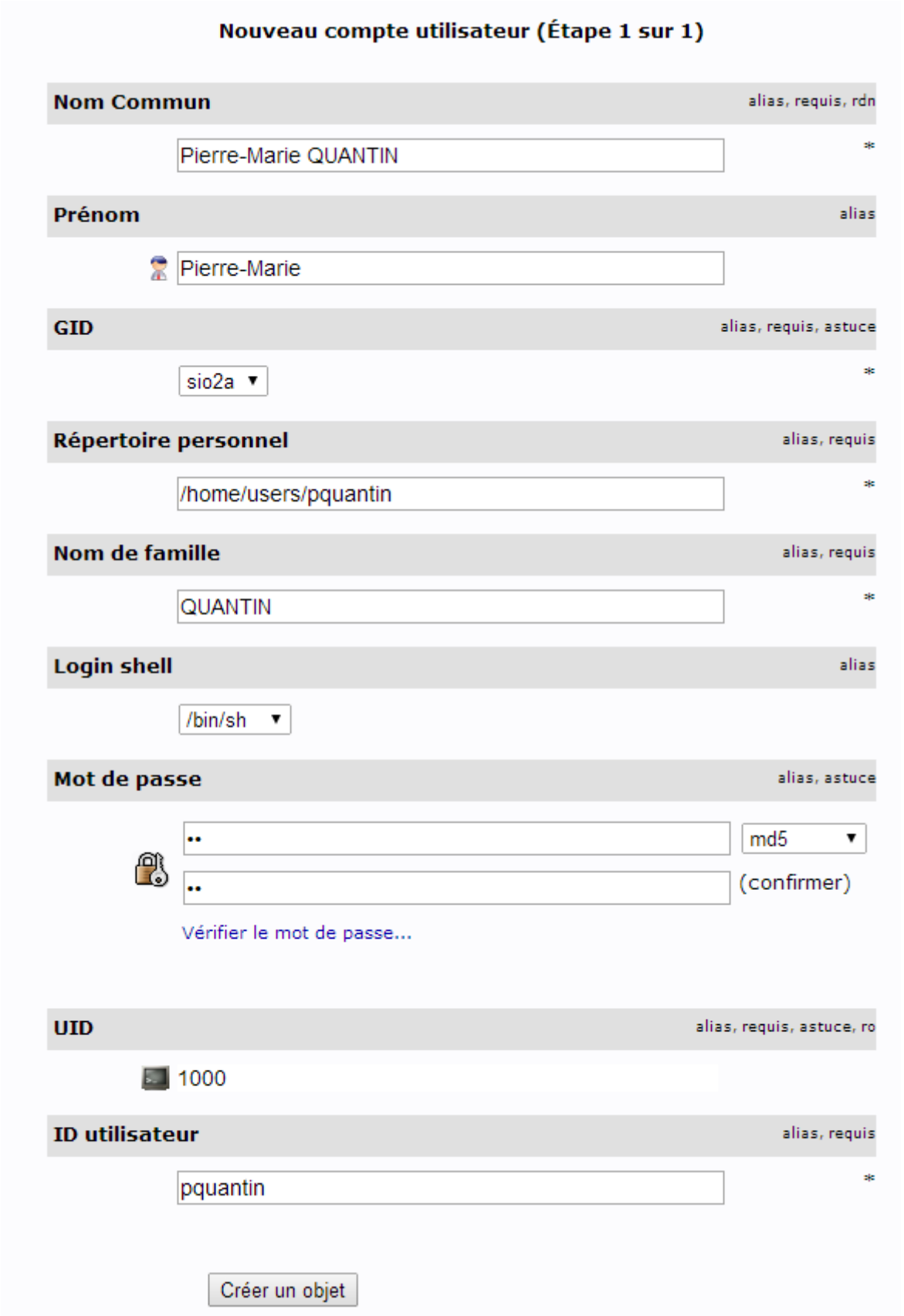
On va maintenant crée une nouvelle entrée « Générique : Unité Organisationnelle » et on valide :



On créer une sous-entrée pour l'UO « etudiant » :



On peut maintenant créer un nouvel utilisateur :



Ainsi, on doit avoir une arborescence de ce type :

On remarque avec la commande *ldapsearch –x* que l'utilisateur est bien crée :

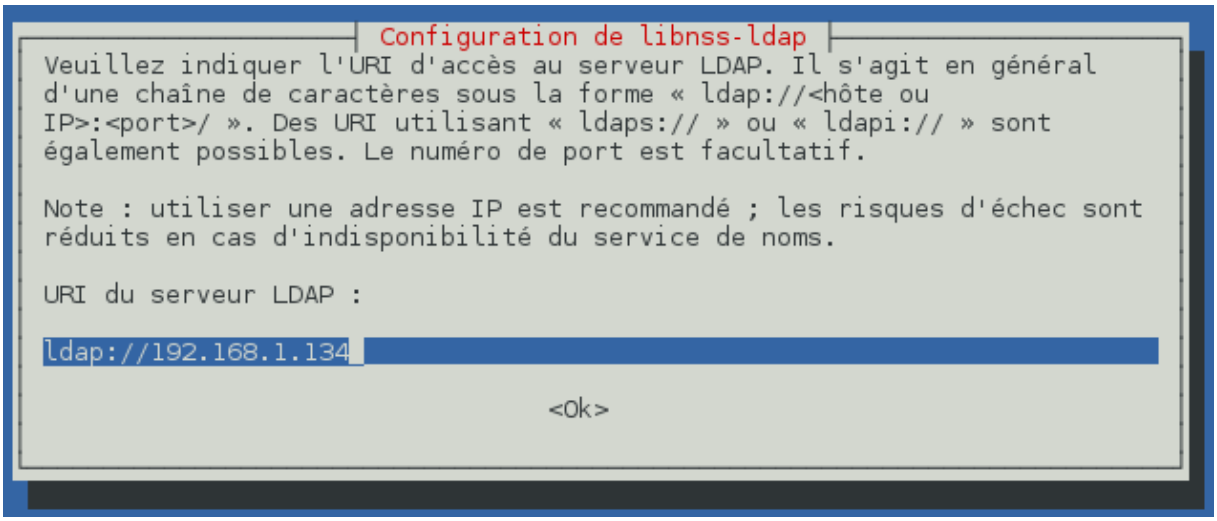
```
# Pierre-Marie QUANTIN, sio2a, etudiant, rezo.local
dn: cn=Pierre-Marie QUANTIN,cn=sio2a,ou=etudiant,dc=rezo,dc=local
cn: Pierre-Marie QUANTIN
givenName: Pierre-Marie
gidNumber: 501
homeDirectory: /home/users/pquantin
sn: QUANTIN
loginShell: /bin/sh
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
uidNumber: 1000
uid: pquantin
```

III. Configuration d'un client Linux pour l'authentification LDAP

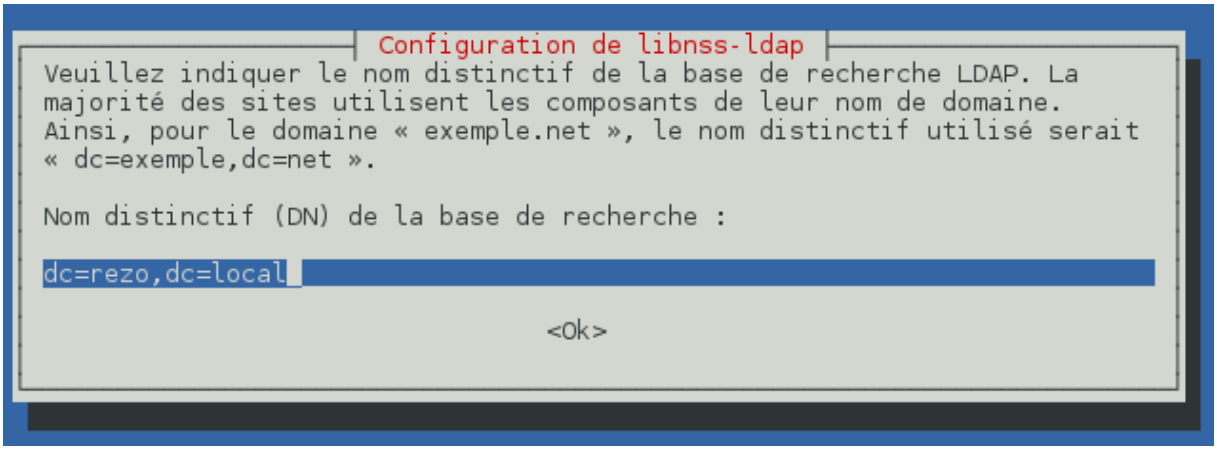
On va installer les paquets nécessaires :

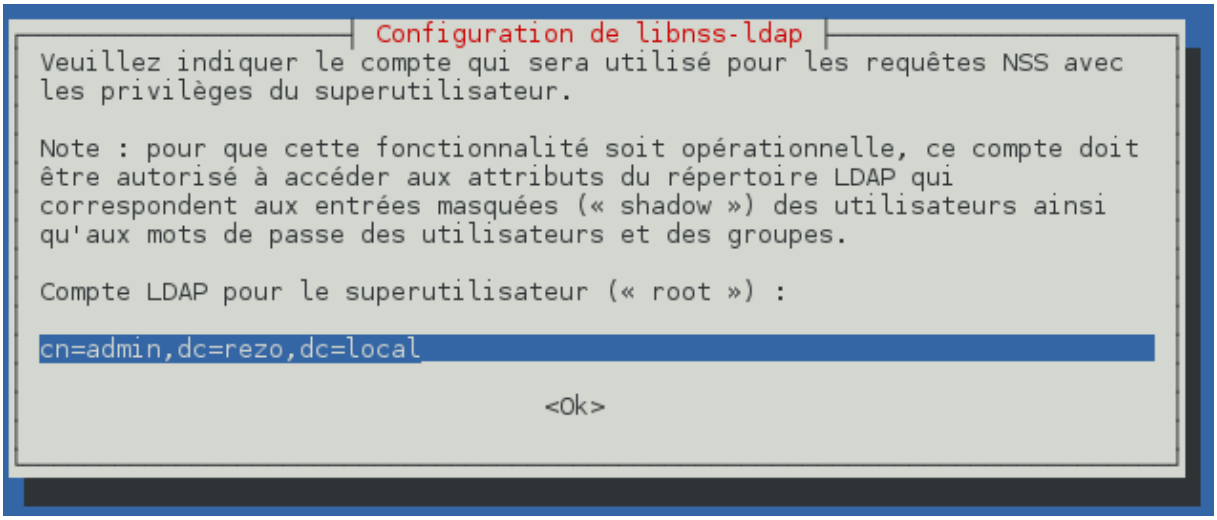
```
root@deb8client:~# apt-get install libnss-ldap libpam-ldap nscd
```

On rentre l'adresse IP du serveur LDAP :



On remplit correctement les DN :





On rentre le mot de passe de l'annuaire LDAP et on répond « Non » aux deux questions.

On va maintenant éditer ce fichier sur le client :

```
root@deb8client:~# nano /etc/ldap/ldap.conf
```

Puis on le modifie par ceci :

```
# See ldap.conf(5) for details
# This file should be world readable but not world writable.

BASE    dc=rezo,dc=local
URI     ldap://192.168.1.134
```

Dernière modification puis on relance le service nscd:

```
  GNU nano 2.2.6          Fichier : /etc/nsswitch.conf

# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the `glibc-doc-reference' and `info' packages installed, try:
# `info libc "Name Service Switch"' for information about this file.

passwd:         compat ldap
group:          compat ldap
shadow:         compat ldap
gshadow:        files

hosts:          files mdns4_minimal [NOTFOUND=return] dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       ldap
```

On va passer à la configuration de PAM. On va éditer 7 fichiers. Voici le premier :

```
  GNU nano 2.2.6          Fichier : /etc/pam.d/common-auth                Modifié

# pam-auth-update(8) for details.

# here are the per-package modules (the "Primary" block)
auth    [success=2 default=ignore]      pam_unix.so nullok_secure try_first_pass
auth    [success=1 default=ignore]      pam_ldap.so use_first_pass
# here's the fallback if no module succeeds
auth    requisite                       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
auth    required                        pam_permit.so
# and here are more per-package modules (the "Additional" block)
# end of pam-auth-update config
```

Le second :

```
  GNU nano 2.2.6           Fichier : /etc/pam.d/common-account

#
# /etc/pam.d/common-account - authorization settings common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of the authorization modules that define
# the central access policy for use on the system.  The default is to
# only deny service to users whose accounts are expired in /etc/shadow.
#
# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules.  See
# pam-auth-update(8) for details.
#

# here are the per-package modules (the "Primary" block)
account [success=2 new_authtok_reqd=done default=ignore]        pam_unix.so
account [success=1 default=ignore]      pam_ldap.so
# here's the fallback if no module succeeds
account requisite                       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
account required                        pam_permit.so
```

Le troisième :

```
  GNU nano 2.2.6               Fichier : /etc/pam.d/common-password

# login.defs.
#
# See the pam_unix manpage for other options.

# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules.  See
# pam-auth-update(8) for details.

# here are the per-package modules (the "Primary" block)
password        [success=2 default=ignore]      pam_unix.so obscure sha512
password        [success=1 user_unknown=ignore default=die]     pam_ldap.so use_authtok try_first_pass
# here's the fallback if no module succeeds
password        requisite                       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password        required                        pam_permit.so
# and here are more per-package modules (the "Additional" block)
password        optional        pam_gnome_keyring.so
# end of pam-auth-update config
```

Le quatrième :

```
  GNU nano 2.2.6               Fichier : /etc/pam.d/common-session                         Modifié


# here are the per-package modules (the "Primary" block)
session [default=1]                     pam_permit.so
# here's the fallback if no module succeeds
session requisite                       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
session required                        pam_permit.so
# and here are more per-package modules (the "Additional" block)
session required        pam_unix.so
session optional                        pam_ldap.so
session optional        pam_systemd.so
# end of pam-auth-update config
session required                        pam_mkhomedir.so
```

Le cinquième :

```
 GNU nano 2.2.6          Fichier : /etc/pam.d/common-session-noninteractive

# here are the per-package modules (the "Primary" block)
session [default=1]                     pam_permit.so
# here's the fallback if no module succeeds
session requisite                       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
session required                        pam_permit.so
# and here are more per-package modules (the "Additional" block)
session required        pam_unix.so
session optional                        pam_ldap.so
# end of pam-auth-update config
```

Le sixième `/etc/libnss-ldap.conf` (dé commentez la ligne):

```
# The distinguished name to bind to the server with
# if the effective user ID is root. Password is
# stored in /etc/libnss-ldap.secret (mode 600)
# Use 'echo -n "mypassword" > /etc/libnss-ldap.secret' instead
# of an editor to create the file.
rootbinddn cn=admin,dc=rezo,dc=local
```

Le dernier fichier `/etc/pam_ldap.conf` il faut faire ces modifications :

```
# The distinguished name to bind to the server with
# if the effective user ID is root. Password is
# stored in /etc/pam_ldap.secret (mode 600)
rootbinddn cn=admin,dc=rezo,dc=local
```

```
# Filter to AND with uid=%s
pam_filter objectclass=posixAccount

# The user ID attribute (defaults to uid)
pam_login_attribute uid
```

On se connecte avec un utilisateur LDAP sur le client et dans le terminal, on vérifie :

```
$ id
uid=1000(quantin) gid=501(sio2a) groupes=501(sio2a)
$ pwd
/home/users/pquantin
```