Table des matières

Avant-Propos

E6 :

Elaboration de documents relatifs à la production et à la fourniture de services
A1.1.1 , Analyse du cahier des charges d'un service à produire
A1.2.4 , Détermination des tests nécessaires à la validation d'un service
A1.3.4 , Déploiement d'un service
A4.1.9 , Rédaction d'une documentation technique

Installation

Pour installer le service, il faut télécharger le paquet suivant :

```
root@ldap:~# wget ftp://ftp.openldap.org/pub/OpenLDAP/openldap-release/openldap-
2.4.44.tgz
```

Une fois le paquet téléchargé, il faut télécharger les paquets suivants avec la commande suivante :

```
root@ldap:~# apt-get install libtool libltdl-dev libssl-dev libdb5.3-dev libsasl
2-dev make
```

Une fois l'installation terminée, nous pouvons extraire le fichier téléchargé précédemment :

```
root@ldap:~# tar xzvf openldap-2.4.44.tgz
```

Une fois que l'on a extrait le fichier il suffit de se rendre dans le dossier qui vient d'être extrait :

```
root@ldap:~/openldap-2.4.44# cd openldap-2.4.44/_
```

Il faut ensuite rentrer la commande suivante pour configurer le service :

```
root@ldap:~# ./configure --enable-crypt=yes --enable-lmpasswd=yes --enable-spas
wd=yes --enable-modules=yes --enable-overlays=yes_
```

Ensuite il faut créer les dépendances avec la commande suivante :

```
root@ldap:~/openldap-2.4.44# make depend_
```

Puis faire la commande suivante :

```
root@ldap:~/openldap-2.4.44# make_
```

Et enfin faire le make install :

```
root@ldap:~/openldap-2.4.44# make install_
```

Pour éviter de faire tourner le serveur autrement qu'avec root, on crée un utilisateur openldap sans shell :

```
root@ldap:~/openldap-2.4.44# useradd -s /bin/false -d /usr/local/var/openldap-da
ta openldap_
```

Configuration de base du serveur

Maintenant nous allons configurer le serveur en modifiant le fichier slapd.conf qui est situé dans /usr/local/etc/openldap/

```
root@ldap:~/openldap-2.4.44# cd /usr/local/etc/openldap/_
```

Nous pouvons donc modifier le fichier de conf suivant :

```
root@ldap:/usr/local/etc/openldap# nano slapd.conf_
```

Il faut le modifier pour qu'il y ait les choses suivantes :

```
#
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
include         /usr/local/etc/openldap/schema/core.schema
include         /usr/local/etc/openldap/schema/cosine.schema
include         /usr/local/etc/openldap/schema/inetorgperson.schema
include         /usr/local/etc/openldap/schema/openldap.schema
include         /usr/local/etc/openldap/schema/nis.schema

# Define global ACLs to disable default read access.

# Do not enable referrals until AFTER you have a working directory
# service AND an understanding of referrals.
#referral       ldap://root.openldap.org

pidfile         /usr/local/var/run/slapd.pid
argsfile        /usr/local/var/run/slapd.args

# Load dynamic backend modules:
# modulepath    /usr/local/libexec/openldap
# moduleload    back_mdb.la
# moduleload    back_ldap.la

# Sample security restrictions
#       Require integrity protection (prevent hijacking)
#       Require 112-bit (3DES or better) encryption for updates
#       Require 63-bit encryption for simple bind
# security ssf=1 update_ssf=112 simple_bind=64

# Sample access control policy:
#       Root DSE: allow anyone to read it
#       Subschema (sub)entry DSE: allow anyone to read it
#       Other DSEs:
#               Allow self write access
#               Allow authenticated users read access
#               Allow anonymous users to authenticate
#       Directives needed to implement policy:
access to dn.base="" by * read
access to dn.base="cn=Subschema" by * read
access to *
        by self write
        by users read
        by anonymous auth
#
# if no access controls are present, the default policy
# allows anyone and everyone to read anything but restricts
# updates to rootdn.  (e.g., "access to * by * read")
#
# rootdn can always read and write EVERYTHING!
```

```
#######################################################################
# MDB database definitions
#######################################################################
database config
rootdn          "cn=manager,cn=config"
rootpw          password

database        bdb
#maxsize                1073741824
suffix          "dc=rezo,dc=com"
rootdn          "cn=admin,dc=rezo,dc=com"
# Cleartext passwords, especially for the rootdn, should
# be avoid.  See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
rootpw          password
# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd and slap tools.
# Mode 700 recommended.
directory       /usr/local/var/openldap-data
# Indices to maintain
index   objectClass     eq
index   uid     eq
index   cn,gn,mail      eq,sub
index   ou eq
index   default eq,sub
```

Nous créons maintenant un administrateur (rootdn) nommé manager avec un mot de passe (password). Le type de base de données à utiliser sera bdb de Berkeley DB. Les fichiers de la base de données doivent se situer dans /usr/local/var/openldap-data. Puis nous la transformerons au format LDIF.

Il faut donc créer le répertoire slapd.d

```
root@ldap:/usr/local/etc/openldap# mkdir slapd.d
```

Il faut ensuite faire la commande suivante, il ne faut pas faire attention au message d'erreur :

```
root@ldap:/usr/local/etc/openldap# slaptest -f slapd.conf -F slapd.d
57fb536d bdb_db_open: warning - no DB_CONFIG file found in directory /usr/local/var/openldap-data: (2).
Expect poor performance for suffix "dc=rezo,dc=com".
57fb536d bdb_db_open: database "dc=rezo,dc=com": db_open(/usr/local/var/openldap-data/id2entry.bdb) failed: No such
 file or directory (2).
57fb536d backend_startup_one (type=bdb, suffix="dc=rezo,dc=com"): bi_db_open failed! (2)
slap_startup failed (test would succeed using the -u switch)
root@ldap:/usr/local/etc/openldap# ls
DB_CONFIG.example  ldap.conf.default  slapd.conf          slapd.conf.save  slapd.ldif
ldap.conf          schema             slapd.conf.default  slapd.d          slapd.ldif.default
```

On change maintenant les droits sur le fichier :

```
root@ldap:/usr/local/etc/openldap# chown -R openldap.openldap /usr/local/etc/openldap
```

Maintenant il faut créer le fichier /usr/local/var/openldap-data/DB_CONFIG que slapd va utiliser pour gérer les bases de type Berkeley DB. Un exemple est fourni.

On va donc renommer le fichier :

```
root@ldap:/usr/local/etc/openldap# mv /usr/local/var/openldap-data/DB_CONFIG.example /usr/local/var/openldap-data/D
B_CONFIG
```

On change maintenant les droits :

```
root@ldap:/usr/local/etc/openldap# chown -R openldap.openldap /usr/local/var/openldap-data
```

```
root@ldap:/usr/local/etc/openldap# /usr/local/libexec/slapd -u openldap -g openldap -h 'ldap:///'
```

Maintenant on peut initialiser le service. Les options -u et -g indiquent sous quel utilisateur et groupe le serveur doit tourner et l'option -h indique le type de connexion supportée (ici connexion simple). Pour passer en mode debug et interdire au serveur de se mettre en arrière-plan :

```
root@ldap:/usr/local/etc/openldap# /usr/local/libexec/slapd -d 3
```

Une fois que la commande est rentrée, nous ne pouvons plus rien faire.

```
57fb56ed bdb_db_open: database "dc=rezo,dc=com": dbenv_open(/usr/local/var/openldap-data).
57fb56ed bdb_monitor_db_open: monitoring disabled; configure monitor database to enable
57fb56ed slapd starting
```

Pleins de choses sont marquées avant, et il ne faut toucher à rien. Pour pouvoir continuer la suite du TP, il faut ouvrir une nouvelle console. Par chance, j'avais utilisé Putty, il suffit de le laisser tourner en arrière-plan et d'utiliser la console de base de VirtualBox.

```
root@ldap:/usr/local/etc/openldap# slapcat -s cn=config | less_
```

Une fois que cette commande est faite, il faut rester appuyé sur entré, et une fois que le message END est marqué il faut appuyer sur la touche Q.

Maintenant on peut se connecter pour lister ce qui est présent dans la base de données avec la commande suivante :

```
root@ldap:/usr/local/etc/openldap# ldapsearch -b cn=config -D "cn=manager,cn=con
fig" -w password_
```

Pleins de choses vont s'afficher, et on peut regarder que dans l'autre console des choses sont apparues aussi.

Injection des données

Pour injecter des données, il faut créer le fichier suivant et le configurer correctement :

```
root@ldap:/usr/local/etc/openldap# nano init.ldif_
```

Il faut faire très attention à la syntaxe du fichier.

```
dn:     dc=rezo,dc=com
objectclass:    dcObject
objectclass:    organization
o:      Linux
dc:     rezo

dn:     cn=admin,dc=rezo,dc=com
objectclass:    organizationalRole
cn:     admin
_
```

Après la dernière ligne, il faut OBLIGATOIREMENT mettre un espace, en gros il faut faire une ligne vide à la fin du fichier tout le temps !!!

Une fois que le fichier est créé il faut utiliser la commande suivante pour ajouter les nouvelles entrées.

```
root@ldap:/usr/local/etc/openldap# ldapadd -x -D "cn=admin,dc=rezo,dc=com" -w pa
ssword -f init.ldif
adding new entry "dc=rezo,dc=com"

adding new entry "cn=admin,dc=rezo,dc=com"
```

On peut d'ailleurs voir sur l'autre console que des choses ce sont écrites.

Pour valider le fait que les entrées ont correctement étés ajoutées, on tape la commande suivante :

```
root@ldap:/usr/local/etc/openldap# ldapsearch -LLL -x -D "cn=admin,dc=rezo,dc=co
m" -w password -b 'dc=rezo,dc=com' '(objectclass=*)'
dn: dc=rezo,dc=com
objectClass: dcObject
objectClass: organization
o: Linux
dc: rezo

dn: cn=admin,dc=rezo,dc=com
objectClass: organizationalRole
cn: admin
```

Même démarche pour les OU de base qui servent à créer les utilisateurs et les groupes (OU utilisateur : peaople, OU groupes : groups). Le fichier s'appelle ou.ldif, il faut donc le créer :

```
dn:     ou=people,dc=rezo,dc=com
objectclass:    organizationalUnit
ou:     people

dn:     ou=groups,dc=rezo,dc=com
objectclass:    organizationalUnit
ou:     groups

_
```

Et il ne faut pas oublier le saut de ligne à la fin !

Pour ajouter les utilisateurs, il faut taper la commande suivante, ce qui va les ajouter.

```
root@ldap:/usr/local/etc/openldap# ldapadd -x -D "cn=admin,dc=rezo,dc=com" -w pa
ssword -f ou.ldif
adding new entry "ou=people,dc=rezo,dc=com"

adding new entry "ou=groups,dc=rezo,dc=com"
```

Pour créer un utilisateur sfonfec, le fichier user.ldif est le suivant :

```
root@ldap:/usr/local/etc/openldap# nano users.ldif_
```

```
dn:     cn=sfonfec,ou=people,dc=rezo,dc=com
objectclass:    top
objectclass:    account
objectclass:    posixAccount
objectclass:    shadowAccount
uid:    sfonfec
uidnumber:      1500
gidnumber:      10000
userpassword:   password
gecos:  Sophie Fonfec
loginshell:     /bin/bash
homedirectory:  /home/sfonfec
shadowwarning:  7
shadowmin:      8
shadowmax:      9999
shadowlastchange:       10877
```

Une fois que le fichier est créé, il suffit de taper la commande suivante afin d'ajouter les données.

```
root@ldap:/usr/local/etc/openldap# ldapadd -x -D "cn=admin,dc=rezo,dc=com" -w pa
ssword -f users.ldif
adding new entry "cn=sfonfec,ou=people,dc=rezo,dc=com"
```

Puis, pour ajouter un groupe, il faut créer le fichier groups.ldif et le remplir.

```
root@ldap:/usr/local/etc/openldap# nano groups.ldif_
```

```
dn:     cn=ldap,ou=groups,dc=rezo,dc=com
objectclass:    top
objectclass:    posixGroup
cn:     ldap
gidNumber:      10000_
```

Une fois que le fichier est créé, il faut rentrer la commande suivante :

```
root@ldap:/usr/local/etc/openldap# ldapadd -x -D "cn=admin,dc=rezo,dc=com" -w pa
ssword -f groups.ldif
adding new entry "cn=ldap,ou=groups,dc=rezo,dc=com"
```

Pour vérifier si les entrées sont correctement ajoutées, il faut faire la commande suivante :

```
root@ldap:/usr/local/etc/openldap# ldapsearch -x -D 'cn=sfonfec,ou=people,dc=rez
o,dc=com' -w password -b 'ou=people,dc=rezo,dc=com' '(cn=sfonfec)' loginshell
# extended LDIF
#
# LDAPv3
# base <ou=people,dc=rezo,dc=com> with scope subtree
# filter: (cn=sfonfec)
# requesting: loginshell
#

# sfonfec, people, rezo.com
dn: cn=sfonfec,ou=people,dc=rezo,dc=com
loginShell: /bin/bash

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

Si la commande nous sort des résultats, alors c'est que les ajouts ont bien été faits.

Remarque : remise à zéro de la base (si on veut tout supprimer, mais il ne faut pas le faire)

-Arrêter le serveur
-Supprimer la configuration de base
#rm -rf /usr/local/etc/openldap/slapd.d/*
-Recréer la configuration au format LDIF et donner les droits
-Pour supprimer les données, purger la base bdb en sauvegardant le fichier DB_CONFIG
#rm -rf /usr/local/var/openldap-data/*
-Et remettre le fichier DB_CONFIG à sa place et affecter le bon propriétaire
-Redémarrer le serveur.

Installation d'un client graphique

phpLDAPadmin est une interface écrite en php qui permet de modifier facilement et via une interface conviviale un annuaire LDAP.

Il faut installer les paquets suivants : Apache2, php5, phpmyadmin.

```
root@ldap:~# apt install apache2_
```

```
root@ldap:~# apt install php5_
```

```
root@ldap:~# apt install phpldapadmin_
```

Pour voir si le service est bien installé, il faut rentrer l'adresse suivante dans un navigateur.

192.168.1.128/phpldapadmin/

Si la page suivante s'affiche, alors c'est que le service fonctionne.

Pour des raisons de sécurités, les droits d'accès sont modifiés, ainsi que le propriétaire.

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

La configuration phpLDAPAdmin nécessite la modification du fichier config.php, situé dans le répertoire /etc/phpldapadmin

La première modification apportée concerne le nom du serveur LDAP qui sera affiché sur l'interface. Le nom affiché par défaut est My LDAP Server. La modification consiste en la modification de la section suivante :

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

La seconde modification concerne la base de recherche, valeur souhaitée dc=rezo,dc=com, dans l'annuaire. Il faut modifier la section suivante :

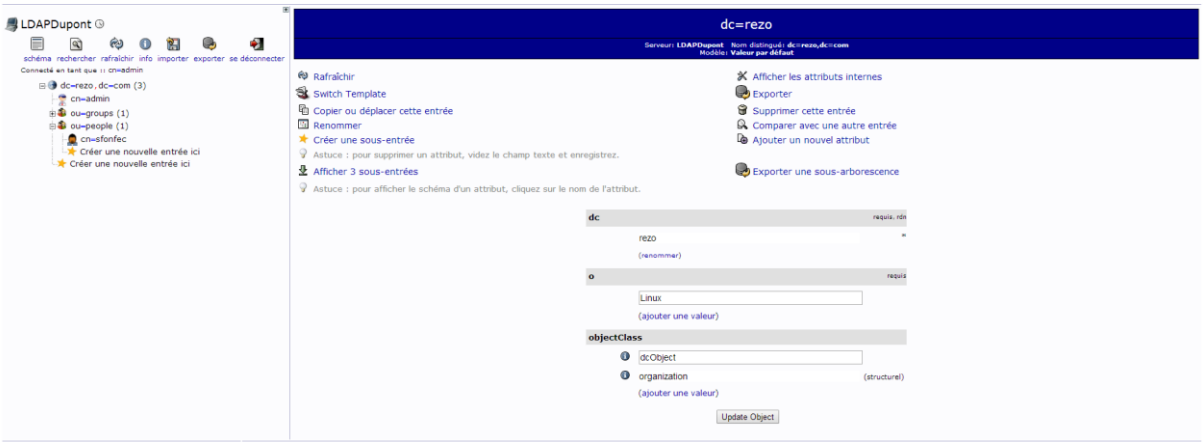
```
/* Array of base DNs of your LDAP server. Leave this blank to have phpLDAPadmin
   auto-detect it for you. */
$servers->setValue('server','base',array('dc=rezo,dc=com'));
```

La troisième modification concerne le compte d'authentification par défaut est cn=admin,dc=exemple,dc=com. Il paraît utile de modifier cette valeur pour être le « vrai » compte administrateur de l'annuaire accédé :

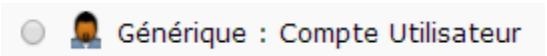
```
/* The DN of the user for phpLDAPadmin to bind with. For anonymous binds or
   'cookie','session' or 'sasl' auth_types, LEAVE THE LOGIN_DN AND LOGIN_PASS
   BLANK. If you specify a login_attr in conjunction with a cookie or session
   auth_type, then you can also specify the bind_id/bind_pass here 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=com');
#  $servers->setValue('login','bind_id','cn=Manager,dc=example,dc=com');
```

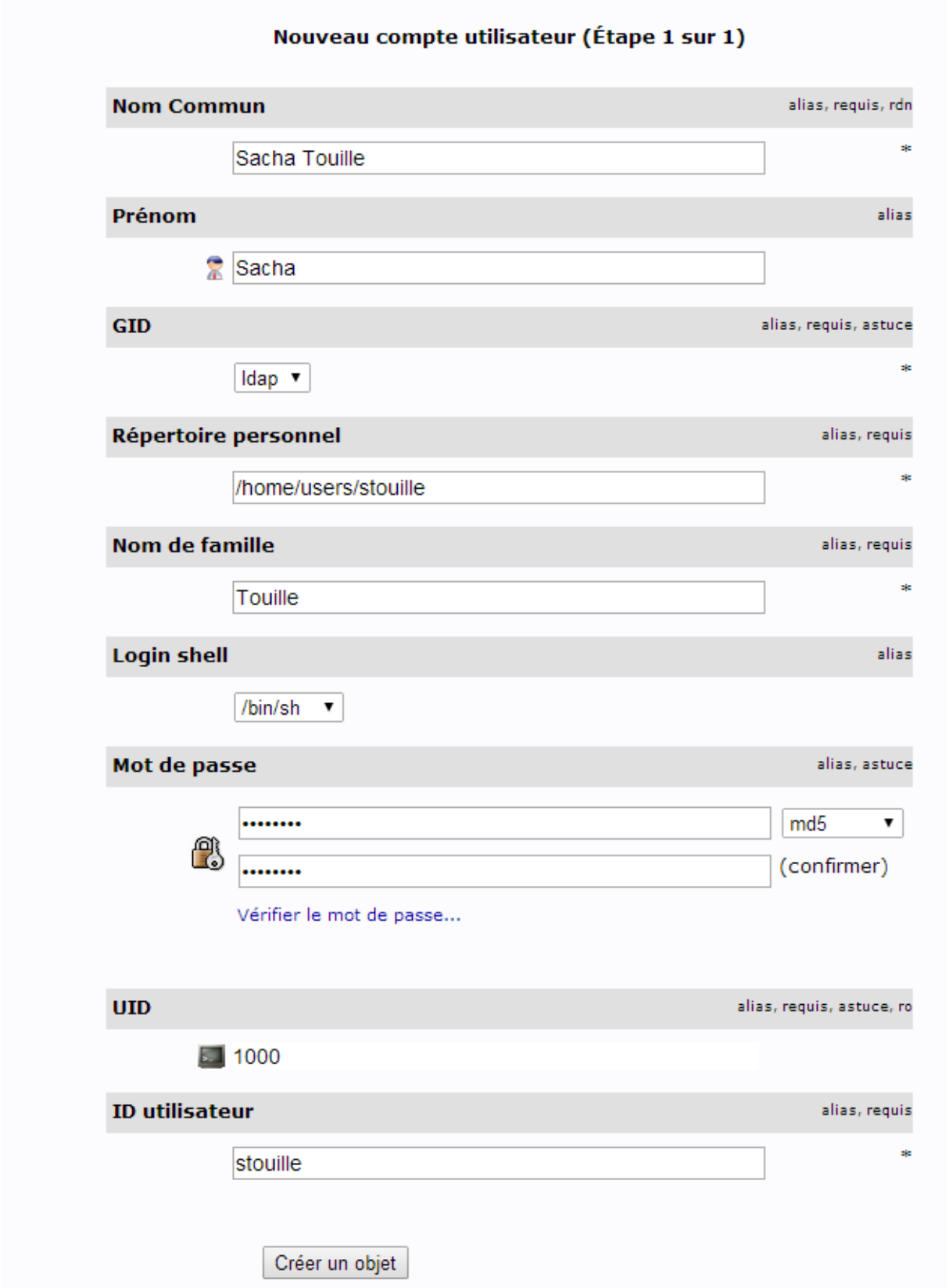
Maintenant, lorsque l'on veut se connecter sur la page web, la fenêtre suivante apparait :

Il faut laisse le DN de connexion tel quel, le mot de passe est celui que l'on a utilisé précédemment, dans notre cas le mot de passe est « password »

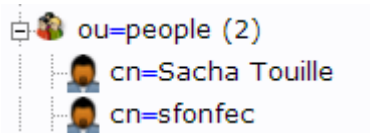


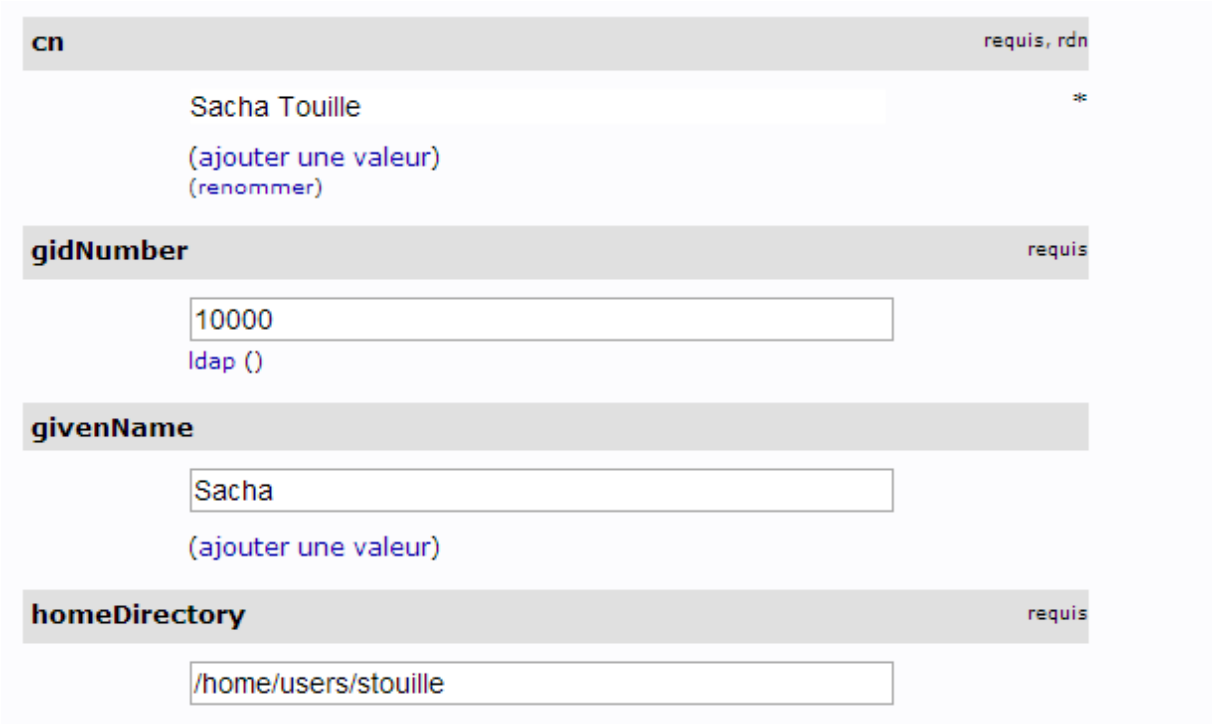
Maintenant nous allons ajouter un nouvel utilisateur dans l'OU people, et voir si il apparait sur le serveur en faisant la commande ldapsearch. Il faut choisir de créer un Compte Utilisateur.





Une fois que l'on a ajouté l'utilisateur, on peut voir qu'il apparait bien dans la liste :





Maintenant, retournons sur le serveur et tapons la commande suivante pour voir si l'utilisateur est bien ajouté :

```
root@ldap:~# ldapsearch -LLL -x -D "cn=admin,dc=rezo,dc=com" -w password -b 'dc=rezo,dc=com' '(objectclass=*)'
```

```
dn: cn=Sacha Touille,ou=people,dc=rezo,dc=com
cn: Sacha Touille
givenName: Sacha
gidNumber: 10000
homeDirectory: /home/users/stouille
sn: Touille
loginShell: /bin/sh
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
userPassword:: e01ENX1YMDNNTzFxblpkWWRneWZ1dUlMUG1RPT0=
uidNumber: 1000
uid: stouille
```

Et voilà ! L'utilisateur est correctement ajouté !

Configuration du serveur LDAP

L'arborescence du répertoire /usr/local/etc/openldap/slap.d/ est :

```
root@ldap:/usr/local/etc/openldap/slapd.d# tree
.
├── cn=config
│   ├── cn=schema
│   │   ├── cn={0}core.ldif
│   │   ├── cn={1}cosine.ldif
│   │   ├── cn={2}inetorgperson.ldif
│   │   ├── cn={3}openldap.ldif
│   │   └── cn={4}nis.ldif
│   ├── cn=schema.ldif
│   ├── olcDatabase={0}config.ldif
│   ├── olcDatabase={1}bdb.ldif
│   └── olcDatabase={-1}frontend.ldif
└── cn=config.ldif

2 directories, 10 files
```

Le DIT est contenu dans le fichier cn=config.ldif qui est généré automatiquement à chaque démarrage du serveur.

Dans le fichier slapd.conf, l'entrée RootDN contient le DN de l'utilisateur autorisé à faire des modifications dans l'annuaire. Son mot de passe est défini par la ligne RootPW. Pour le modifier, il faut faire la commande suivante :

Ldapmodify -h ldap:/// dn :olcDatabase={2}bdb,cn=config changetype: modify replace: olcRootDN olcRootDN: cn=admin,dc=rezo,dc=com add: olcRootPW olcRootPW: siosisr

Cependant !!! Cette commande ne peut marcher que si le mot de passe est chiffré, ce qui n'est pas le cas dans notre TP, la commande ne marche donc pas.