Nom	Prénom	Distribution	Version
Divaret	Nathan	Debian 8.5	1.0

SERVEUR LDAP DEBIAN

SOMMAIRE

Contenu

Nom	Prénom	Distribution	Version
Divaret	Nathan	Debian 8.5	1.0

1) Objectif

Dans cette procédure, nous allons montrer comment installer et configurer un serveur d'annuaire **LDAP** sous Debian 8.5 avec installation des paquets et configuration des fichiers pour le bon fonctionnement du serveur. Cette procédure a été faite dans le cadre d'un TP.

2) Prérequis

Pour réaliser cette procédure, nous avons besoin des éléments suivants :

➔ Debian Jessie 8.5

Nom du serveur LDAP	Adresse IP du serveur LDAP
LDAP	192.168.1.132 /24

3) Définitions

- **LDAP** (**L**ightweight **D**irectory **A**ccess **P**rotocol) est un protocole qui permet d'interroger et de modifier des services d'annuaire. Il représente une norme pour les systèmes d'annuaire locaux, incluant un modèle de données, nommage, sécurité et réplication.
- Un annuaire est un référentiel partagé de personnes et de ressources dont la vocation est de localiser à l'aide de fonctions élaborées de navigation et de recherche et d'offrir des mécanismes de sécurité pour protéger ces informations et y accéder.

4) Installation et configuration du service LDAP sous Debian

Pour commencer, nous mettons à jour les paquets avec la commande :

```
root@LDAP:~# apt-get update
```

Nous devons télécharger l'archive **LDAP avec un WGET** :

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

Nous installation les paquets suivant :

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

Puis nous décompression l'archive :

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

Nous nous rendons dans le dossier « **openldap-2.4.44.tgz** » :

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

Nous allons configurer le fichier :

```
root@LDAP:~/openldap-2.4.44# ./configure --enable-crypt=yes --enable-lmpasswd=ye
s --enable-spasswd=yes --enable-modules=yes --enable-overlays=yes_
```

Nous créons les dépendances et mettons en relation les fichiers :

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

Nous compilons les fichiers :

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

Nous installons le programme de compilation des fichiers :

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

Nous ajoutons un utilisateur nommé « **openldap** » sans shell afin d'éviter de faire fonctionner le serveur autrement qu'avec « **root** » :

```
root@LDAP:~# useradd -s /bin/false -d /usr/local/var/openldap-data/ openldap
root@LDAP:~# _
```

Nous nous rendons dans le fichier « **/usr/local/etc/openldap/slapd.conf** » et le configurons comme suit :

Ces lignes permettent d'insérer les schémas souhaités de l'annuaire **LDAP** :

```
  GNU nano 2.2.6     Fichier : /usr/local/etc/openldap/slapd.conf

#
# 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.
```

Celles-ci permettent le droit de lecture de la base de données (BDD) et sa sous-arborescence. Le propriétaire (l'utilisateur) peut modifier seulement ses propres données, un utilisateur déjà connecté peut tout lire et les autres ont le droit de s'authentifier.

```
#       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
#
```

Ces lignes définissent le choix du format de BDD pour le stockage des données, l'administrateur de l'annuaire et son mot de passe, la configuration de la racine de l'annuaire en fonction du domaine **DNS** :

```
database        config
rootdn          "cn=manager,cn=config"
rootpw          password

database        bdb
suffix          "dc=rezo,dc=com"
rootdn          "cn=admin,dc=rezo,dc=com"

# Cleartext passwords, especially for the
# be avoid.  See slappasswd(8) and slapd.c
# Use of strong authentication encouraged.
rootpw          password
```

Ici, nous précisons la création des index comme pour une BDD incluant les données « **c**ommo**n**name (**cn**) », « **g**ive**n**ame (**gn**) » et « **mail** » (Email) : L'index « **eq** » permet de définir des requêtes comme un nom d'utilisateur ou un nombre et « **sub** » correspond aux requêtes de type **s**ur**n**ame (**sn**) :

```
# Indices to maintain
index   objectClass     eq
index   uid     eq
index   cn,gn,mail      eq,sub
index   ou eq
index   default eq,sub
```

Nous créons un nouveau dossier nommé « **slapd.d** » dans le répertoire « **/usr/local/etc/openldap** » et nous nous y rendons :

```
root@LDAP:~# mkdir /usr/local/etc/openldap/slapd.d
root@LDAP:~# _
```

Nous nous rendons dans le répertoire « **/usr/local/etc/openldap** » et essayons de convertir le fichier « **slapd.conf** » et nous avons une plainte que nous ignorons pour le moment :

```
root@LDAP:/usr/local/etc/openldap# slaptest -f slapd.conf -F slapd.d
57fb58d8 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".
57fb58d8 bdb_db_open: database "dc=rezo,dc=com": db_open(/usr/local/var/openldap
-data/id2entry.bdb) failed: No such file or directory (2).
57fb58d8 backend_startup_one (type=bdb, suffix="dc=rezo,dc=com"): bi_db_open fai
led! (2)
slap_startup failed (test would succeed using the -u switch)
root@LDAP:/usr/local/etc/openldap# _
```

Nous modifions les permissions d'accès pour « **openldap** » du répertoire « **/usr/local/etc/openldap** » avec un **chown** :

```
root@LDAP:/usr/local/etc/openldap# chown -R openldap.openldap /usr/local/etc/ope
nldap/
root@LDAP:/usr/local/etc/openldap# _
```

Nous créons un nouveau fichier nommé « **DB_CONFIG** » :

```
root@LDAP:~# mv /usr/local/var/openldap-data/DB_CONFIG.example /usr/local/var/op
enldap-data/DB_CONFIG
root@LDAP:~# _
```

Nous modifions les droits d'accès pour « **openldap** » :

```
root@LDAP:~# chown -R openldap.openldap /usr/local/var/openldap-data/
root@LDAP:~# _
```

Nous allons tester la connexion **LDAP** :

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

« **-u** » et « **-g** » indiquent sous quel utilisateur et groupe le serveur doit tourner.

Nous interdisons au serveur de mettre en arrière-plan :

```
root@LDAP:~# /usr/local/libexec/slapd -d 3
```

```
57fb6244 slapd startup: initiated.
57fb6244 backend_startup_one: starting "cn=config"
57fb6244 config_back_db_open
57fb6244 backend_startup_one: starting "dc=rezo,dc
57fb6244 bdb_db_open: database "dc=rezo,dc=com": d
dap-data).
57fb6244 bdb_monitor_db_open: monitoring disabled;
 enable
57fb6244 slapd starting
```

Nous ouvrons une nouvelle console avec l'utilitaire « **Putty** » pour administrer plus simplement:

```
root@LDAP:~# slapcat -s cn=config | less
```

Arrivé ici, nous tapons « **q** » pour quitter :

```
olcObjectIdentifier: OLcfgCtOc OLcfgOc:4
olcObjectIdentifier: OMsyn 1.3.6.1.4.1.1466.115.121.1
olcObjectIdentifier: OMsBoolean OMsyn:7
olcObjectIdentifier: OMsDN OMsyn:12
olcObjectIdentifier: OMsDirectoryString OMsyn:15
olcObjectIdentifier: OMsIA5String OMsyn:26
:
```

Nous testons le serveur **LDAP** avec le mot de passe :

```
root@LDAP:~# ldapsearch -b cn=config -D "cn=manager,cn=config" -w password
```

```
# search result
search: 2
result: 0 Success

# numResponses: 11
# numEntries: 10
root@LDAP:~#
```

Tout cela fonctionne correctement.

5) Injection des données

Les fichiers avec l'extension « **.ldif** » sont indispensables pour insérer des données.

Une fois dans le dossier « **/usr/local/etc/openldap** », nous créons le fichier « **init.ldif** » pour l'insertion des données et saisissons le contenu suivant :

```
  GNU nano 2.2.6                Fichier : /usr/local/etc/openldap/init.ldif

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

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

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

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

root@LDAP:/usr/local/etc/openldap#
```

Pour valider les 2 champs ci-dessus (copie d'écran), nous validons de la manière suivante :

```
root@LDAP:/usr/local/etc/openldap# ldapsearch -LLL -x -D "cn=admin,dc=rezo,dc=com" -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

root@LDAP:/usr/local/etc/openldap#
```

Maintenant, nous créons le fichier « **ou.ldif** » pour la création d'utilisateurs et de groupes et leur classe d'objet et saisissons le contenu suivant :

```
  GNU nano 2.2.6                Fichier : /usr/local/etc/openldap/ou.ldif

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

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

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

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

Ensuite, nous créons un nouveau fichier nommé « **users.ldif** » pour la création d'un utilisateur nommé « **sfonfec** » :

```
  GNU nano 2.2.6                Fichier : /usr/local/etc/openldap/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/fash
homedirectory:          /home/sfonfec
shadowwarning:          7
shadowmin:      8
shadowmax:      9999
shadowlastchange:       10877
```

Enfin, nous créons le fichier « **groups.ldif** » pour la création de groupes et saisissons le contenu suivant :

```
  GNU nano 2.2.6                Fichier : /usr/local/etc/openldap/groups.ldif

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

Nous insérons ces 2 nouveaux fichiers au serveur **LDAP** et constatons que ceux-ci sont bien insérés via l'élément « **adding new entry** » :

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

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

-

-

Nous vérifions la configuration de ces nouveaux fichiers via la commande suivante :

```
root@LDAP:/usr/local/etc/openldap# ldapsearch -x -D 'cn=sfonfec,ou=people,dc=rezo,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/fash

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
root@LDAP:/usr/local/etc/openldap#
```

Cette commande permet de se connecter avec le compte utilisateur « **sfonfec** » et de récupérer correctement un paramètre de son compte.

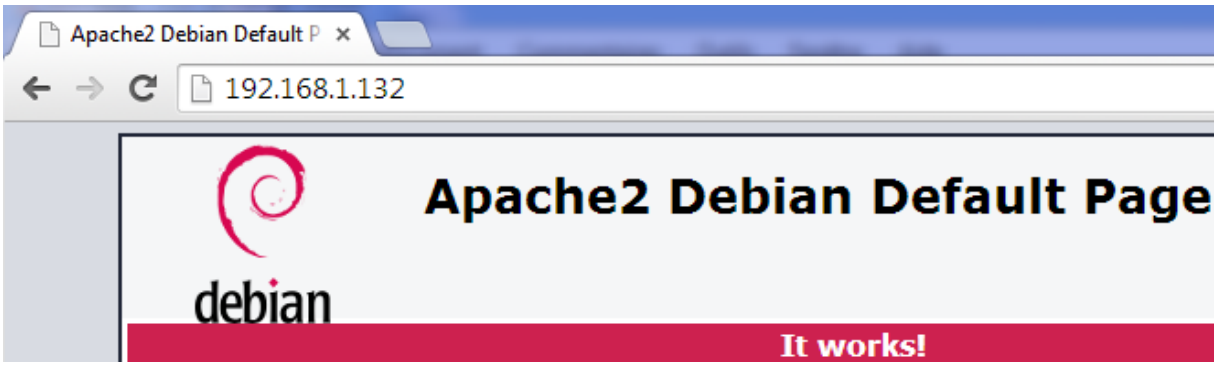
6) Installation et configuration d'un client graphique

Installation du client

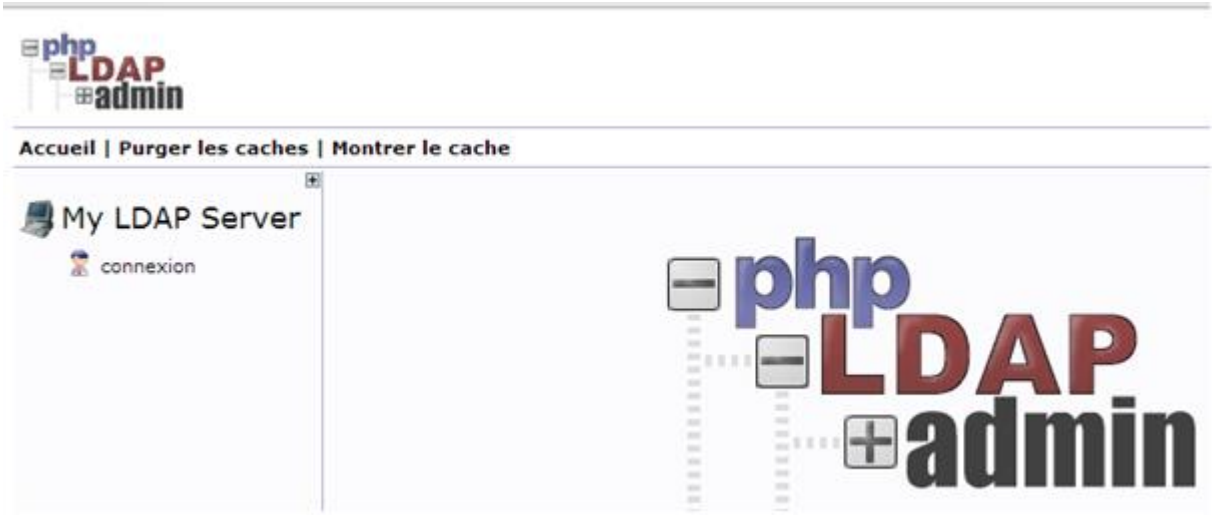
En premier, nous installons les paquets « **apache2** », « **php5** » et « **phpldapadmin** » :

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

Nous testons le service « **apache2** » :



Et, nous pouvons accéder à l'interface Web de **LDAP** :



Pour des raisons de sécurité, nous devons modifier les droits d'accès 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/
root@LDAP:~#
```

Nous nous rendons dans le fichier « **/etc/phpldapadmin/config.php** » et procédons aux modifications suivantes :

Première modification : Le nom du serveur sera affiché sur le serveur :

```
/* A convenient name that will appear in the tree viewer and throughout
   phpLDAPadmin to identify this LDAP server to users. */
$servers->setValue('server','name','192.168.1.132');
```

Seconde modification : La base de recherche sera présente dans l'annuaire :

```
/* 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'));
```

Troisième modification : Le compte d'authentification sera affiché par défaut :

```
/* 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');
```

Si nous le souhaitons, nous pouvons également modifier le « **timezone** » afin que l'interface Web soit en français :

```
/* 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';
```

7) Tests de fonctionnement du serveur LDAP

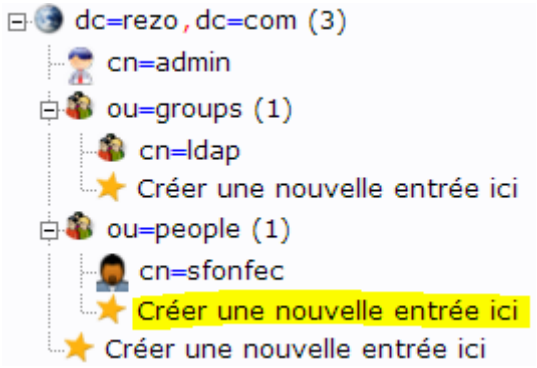
a) Tests de connexion de l'utilisateur « **root** »

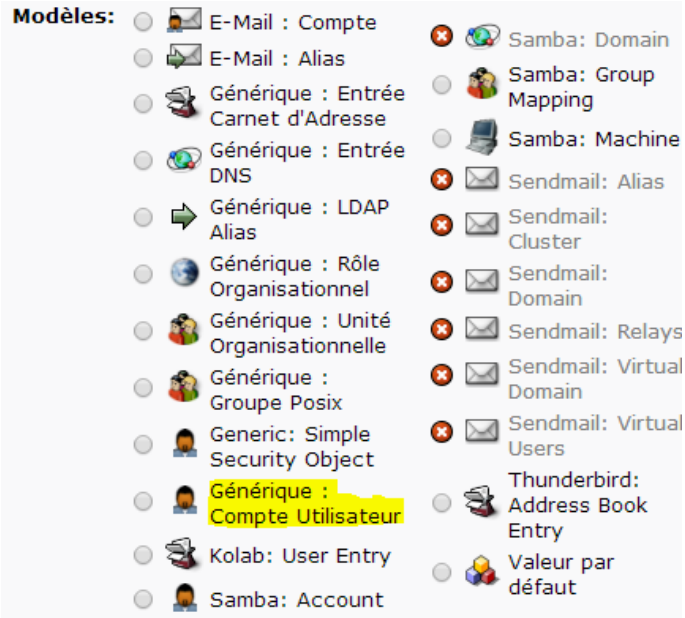
- Nous nous rendons sur l'interface Web du serveur **LDAP** pour vérifier le bon fonctionnement en nous connectant :

- Nous saisissons le mot de passe du serveur qui est « **password** » (Ce mot de passe « **root** » attribué est visible dans le fichier « **/usr/local/etc/openldap/slapd.conf** ») :

Nous constatons que nous sommes bien connectés au serveur :

Pour tester, nous allons ajouter un utilisateur dans « **ou=people** » (déjà créé) :





Nous remplissons tous les champs pour ce nouvel utilisateur et créons cet objet :

Nous vérifions et constatons que cet utilisateur est bien présent :

Nous allons également vérifier sa présence via cette commande :

```
ldapsearch -x -D "cn=admin,dc=rezo,dc=com" -w password -b 'ou=people,dc=rezo,dc=com' '(objectclass=*)'
```

8) Configuration du serveur LDAP

Nous allons visualiser l'arborescence du dossier « **/usr/local/etc/openldap/slapd.d** ». Pour ce faire, nous installons le paquet « **tree** » :

```
root@LDAP:~# apt-get install tree
```

Et, nous visualisons cette dernière :

```
root@LDAP:~# tree /usr/local/etc/openldap/slapd.d/
/usr/local/etc/openldap/slapd.d/
├── 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
```

Un annuaire **LDAP** est un annuaire électronique, composé d'un ou plusieurs arbres de données qui centralisent les informations de l'entreprise. Cette structure hiérarchique est appelé **DIT** (**D**irectory **I**nformation **T**ree).

Nous consultons le contenu du **DIT** dans le fichier « **cn=config.ldif** » et constatons que les informations sont bien répertoriées :

```
  GNU nano 2.2.6            Fichier : /usr/local/etc/openldap/slapd.d/cn=config.ldif

# AUTO-GENERATED FILE - DO NOT EDIT!! Use ldapmodify.
# CRC32 aeeab9b9
dn: cn=config
objectClass: olcGlobal
cn: config
olcConfigFile: slapd.conf
olcConfigDir: slapd.d
olcArgsFile: /usr/local/var/run/slapd.args
olcAttributeOptions: lang-
olcAuthzPolicy: none
olcConcurrency: 0
olcConnMaxPending: 100
olcConnMaxPendingAuth: 1000
olcGentleHUP: FALSE
olcIdleTimeout: 0
olcIndexSubstrIfMaxLen: 4
olcIndexSubstrIfMinLen: 2
olcIndexSubstrAnyLen: 4
olcIndexSubstrAnyStep: 2
olcIndexIntLen: 4
olcListenerThreads: 1
olcLocalSSF: 71
olcLogLevel: 0
olcPidFile: /usr/local/var/run/slapd.pid
olcReadOnly: FALSE
olcSaslSecProps: noplain,noanonymous
olcSockbufMaxIncoming: 262143
olcSockbufMaxIncomingAuth: 16777215
olcThreads: 16
olcTLSCRLCheck: none
olcTLSVerifyClient: never
olcTLSProtocolMin: 0.0
olcToolThreads: 1
olcWriteTimeout: 0
structuralObjectClass: olcGlobal
entryUUID: d62b69c8-2313-1036-8d34-375a8b2c1aea
creatorsName: cn=config
createTimestamp: 20161010090112Z
entryCSN: 20161010090112.189523Z#000000#000#000000
modifiersName: cn=config
modifyTimestamp: 20161010090112Z
```