

# UNIVERSITÉ CHEIKH ANTA DIOP DE DAKAR

École Supérieure Polytechnique



ECOLE SUPERIEURE  
POLYTECHNIQUE

## Rapport Asterisk et Ldap

Présenté par :

Salif BIAYE

Ndeye Astou DIAGOURAGA

Sous la direction de :

Dr Keba

*Enseignant*

Année universitaire 2024-2025

## Table des Matières

---

### I- Installation des Services

---

#### I-a Installation d'Asterisk

---

#### I-b Installation de L'annuaire LDAP

---

### II- Couplage Asterisk et LDAP

---

#### Configuration de l'annuaire LDAP

---

#### II-a Connexion au serveur LDAP

---

#### II-b Configuration générale des comptes SIP

---

#### II-c Configuration générale du dialplan

---

#### Vérification de la connexion au LDAP

---

#### Ajout des utilisateurs

---

#### Test de la configuration

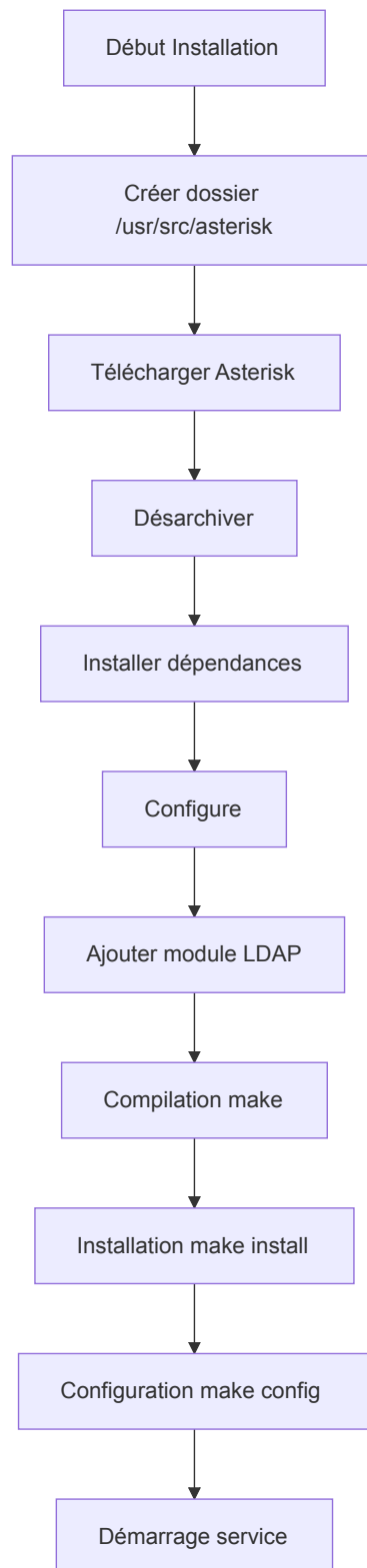
---

# I- Installation des Services

---

## I-a Installation d'Asterisk

---



Pour installer Asterisk, nous allons suivre une série d'étapes précises. Commençons par créer le dossier qui contiendra notre paquet Asterisk.

```
root@asterisk-ldap: /usr/src/asterisk# mkdir /usr/src/asterisk
```

On se déplace dans le dossier cree et on telecharge le paquet asterisk comme suit

```
root@asterisk-ldap: /usr/src/asterisk# cd /usr/src/asterisk
root@asterisk-ldap: /usr/src/asterisk# wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-17-current.tar.gz
--2020-08-25 17:16:34-- http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-17-current.tar.gz
Résolution de downloads.asterisk.org (downloads.asterisk.org)... 76.164.171.238, 2001:
Connexion à downloads.asterisk.org (downloads.asterisk.org)|76.164.171.238|:80... connecté
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : 43455574 (41M) [application/x-gzip]
Enregistre : «asterisk-17-current.tar.gz»

-17-current.tar.gz 98%[=====>] 40,82M 49,5KB/s tps 13s
```

Maintenant on peut désarchiver le fichier tar comme suit

```
root@asterisk-ldap: /usr/src/asterisk# tar -xvzf asterisk-17-current.tar.gz
asterisk-17.6.0/
asterisk-17.6.0/.cleancount
asterisk-17.6.0/.gitignore
asterisk-17.6.0/.gitreview
asterisk-17.6.0/.lastclean
asterisk-17.6.0/.version
asterisk-17.6.0/BSDmakefile
asterisk-17.6.0/BUGS
```

Après avoir désarchivé on se déplace dans le repertoire de base de asterisk selon la version

```
root@asterisk-ldap: /usr/src/asterisk# ls
asterisk-17.6.0  asterisk-17-current.tar.gz

root@asterisk-ldap: /usr/src/asterisk# cd asterisk-17.6.0/
```

Il faut ensuite installer les dépendances requises pour la compilation d'asterisk:

```
root@asterisk-ldap: /usr/src/asterisk/asterisk-17.6.0# apt-get install gcc g++ make
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
gcc-7 gcc-7-base libasan4 libatomic1 libc-dev-bin libc6-dev libcilkrts5
libgcc-7-dev libitm1 liblsan0 libmpx2 libquadmath0 libstdc++-7-dev
libtinfo-dev libtsan0 libubsan0 linux-libc-dev manpages-dev
```

on télécharge le deuxieme

```
root@asterisk-ldap: /usr/src/asterisk/asterisk-17.6.0# ./contrib/scripts/install_prereqs
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
```

```
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  aptitude-common libcwidget3v5
Paquets suggérés :
```

Nous allons ensuite compiler asterisk

```
root@asterisk-ldap: /usr/src/asterisk/asterisk-17.6.0# ./configure
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
```

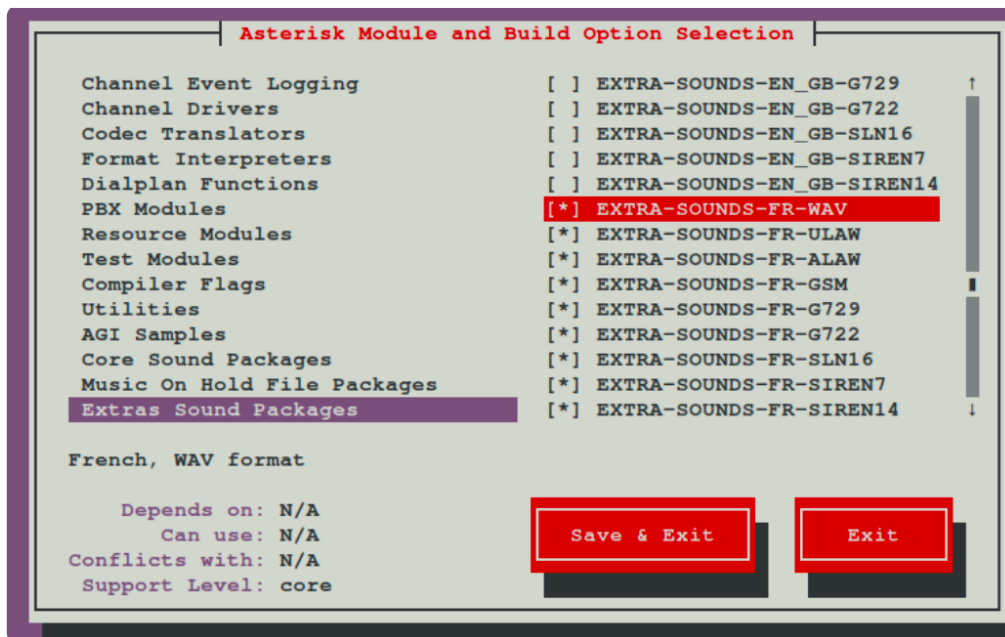
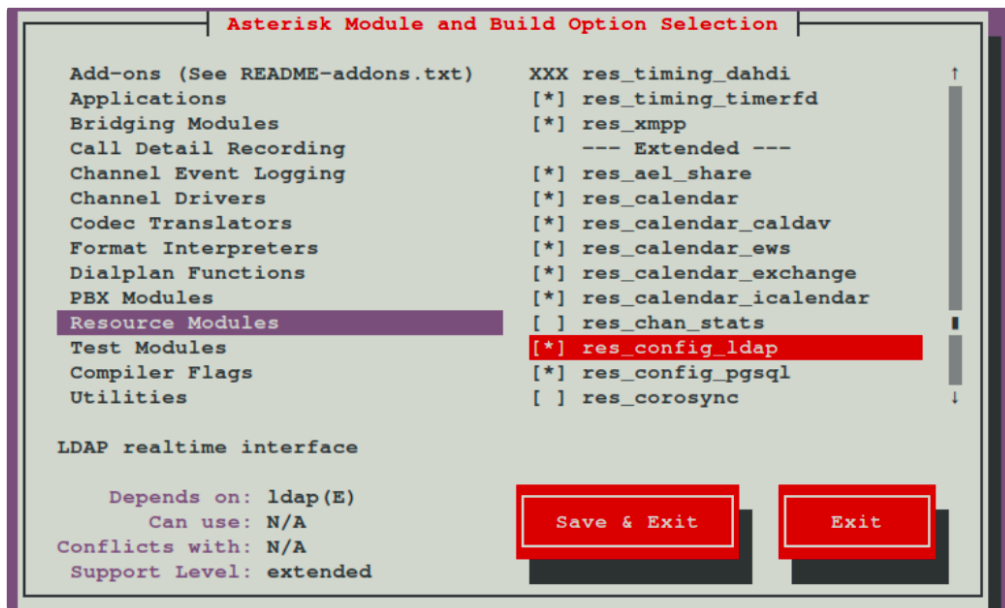
On aura comme résultat

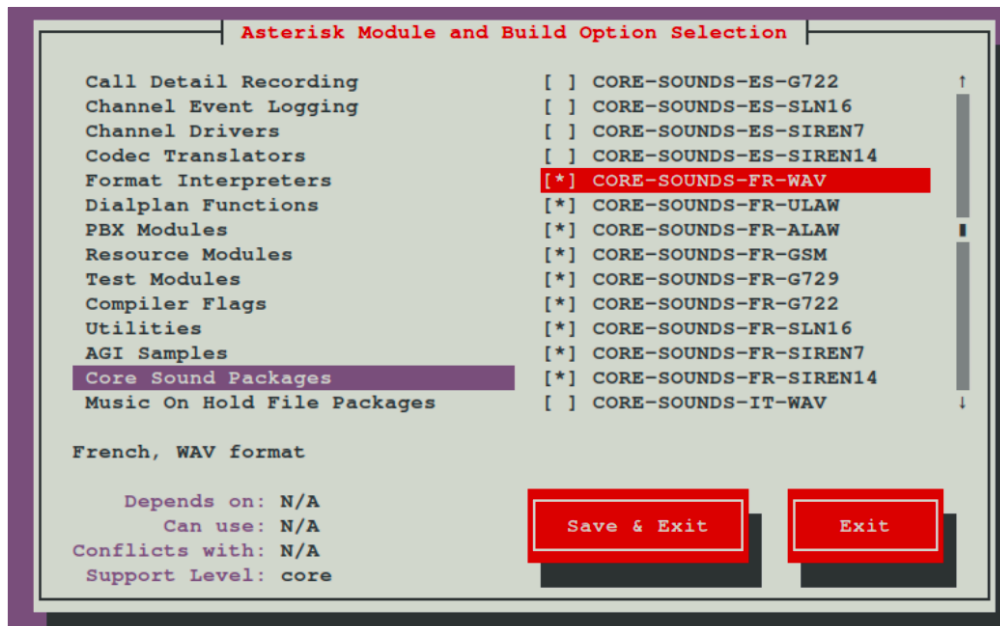
```
-$$$$$$$$$$$$$$=..
      .7$7..      .7$7..
    .$.:.      .7.7
      .7      7$$$$ .677
    ..$.      $$$$ .$$7
    ..7$ .?.      $$$$ .?. 7$$$
    $. .$$7. $$$7 .7$$$ .$$$
    .777. .$$$$$77$77$$$$$7. $$$
    $$$~ .7$$$$$$$$$$$7. .$$$
    .7$ .7$$$$$$$7: ?$$$
    $$$ ?7$$$$$$$I .$$$7
    $$$ .7$$$$$$$$$$$ .$$$
    $$$ $$$$$7$$$$$$$$$$$ .$$$
    $$$ $$$ 7$$$7 .$$$ .$$$
    $$$ $$$$7 .$$$
    7$$$7 7$$$ 7$$$
    $$$$ $$$
    $$$7. $$$ (TM)
    $$$$.. .7$$$$$ $$
    $$$$$$$$7$$$$$$$.$$$$
    .$$$$$$$$$$$$.

# Configuration Asterisk
configure: Package configuré pour:
configure: OS type : linux-gnu
configure: Host CPU : x86_64
configure: build-cpu:vendor:os: x86_64 : pc : linux-gnu :
configure: host-cpu:vendor:os: x86_64 : pc : linux-gnu :
salif@asterisk-ldap:/usr/src/asterisk/asterisk-17.6.0#
```

On ajoute le module LDAP pour Asterisk et sélectionnons des voix francaises:

```
root@asterisk-ldap: /usr/src/asterisk/asterisk-17.6.0# make menuconfig
```





Après tabulation on Save & Exit pour sauvegardé

## Compilation

```
root@asterisk-ldap: /usr/src/asterisk/asterisk-17.6.0# make
[CC] astcanary.c -> astcanary.o
[LD] astcanary.o -> astcanary
[CC] astdb2sqlite3.c -> astdb2sqlite3.o
[CC] hash/hash.c -> hash/hash.o
[CC] hash/hash_bigkey.c -> hash/hash_bigkey.o
[CC] hash/hash_buf.c -> hash/hash_buf.o
[CC] hash/hash_func.c -> hash/hash_func.o
```

## Installation

```
root@asterisk-ldap:/usr/src/asterisk/asterisk-17.6.0# make install
Installing modules from channels...
Installing modules from pbx...
Installing modules from apps...
Installing modules from codecs...
Installing modules from formats...
Installing modules from cdr...
```

```
root@asterisk-ldap:/usr/src/asterisk/asterisk-17.6.0# make samples
Installing adsi config files...
/usr/bin/install -c -d "/etc/asterisk"
Installing configs/samples/asterisk.adsi
Installing configs/samples/telcordia-1.adsi
Installing other config files...
Installing file configs/samples/acl.conf.sample
Installing file configs/samples/adsi.conf.sample
```

```
Installing file configs/samples/agents.conf.sample
Installing file configs/samples/alarmreceiver.conf.sample
Installing file configs/samples/alsa.conf.sample
Installing file configs/samples/amd.conf.sample
```

```
root@asterisk-ldap:/usr/src/asterisk/asterisk-17.6.0# make config
```

on démarre le service asterisk comme suit

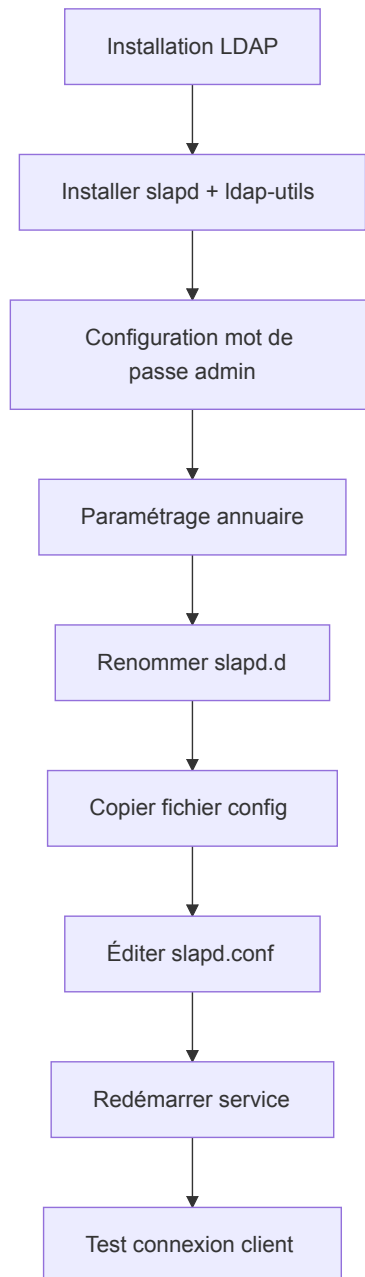
```
root@asterisk-ldap:/usr/src/asterisk/asterisk-17.6.0# /etc/init.d/asterisk start
[ ok ] Starting asterisk (via systemctl): asterisk.service.
root@asterisk-ldap:/usr/src/asterisk/asterisk-17.6.0#
```

Pour vérifier le fonctionnement du serveur Asterisk, on peut lancer la console :

```
root@asterisk-ldap:/usr/src/asterisk/asterisk-17.6.0# asterisk -rvvvvv
Asterisk 17.6.0, Copyright (C) 1999 - 2018, Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public License
=====
Connected to Asterisk 17.6.0 currently running on asterisk-ldap (pid = 16493)
asterisk-ldap:*CLI>
```



## I-b Installation de L'annuaire LDAP



Un annuaire LDAP est une base de donnée non **Relationnelle** .

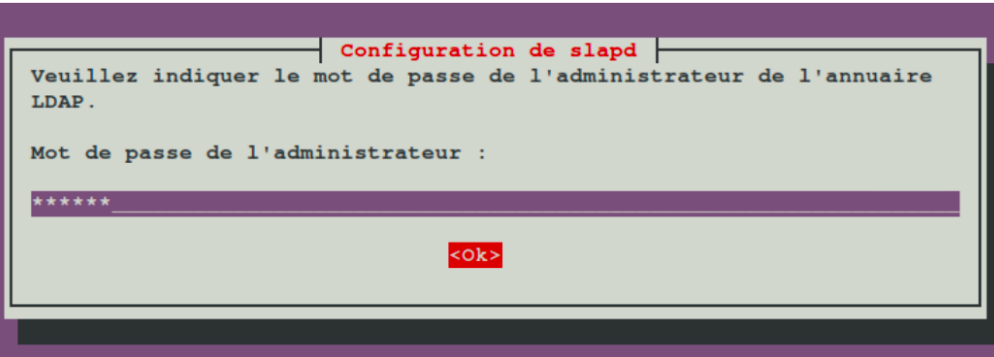
Elle peut être **associé à un système de stockage de données permettant de rendre accessible un ensemble d'informations à tous les utilisateurs de ce système.**

Installation des paquets nécessaires : slapd le server LDAP ldap-utils pour les commandes coté client

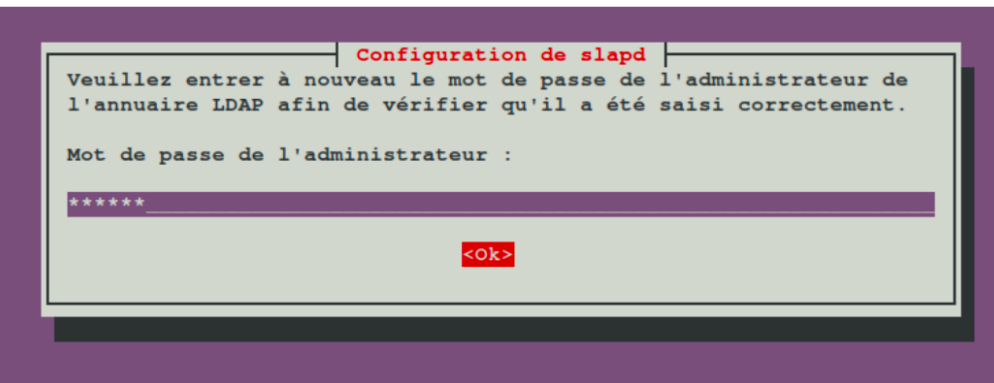
```
root@asterisk-ldap:~# apt install slapd ldap-utils
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
```

```
Lecture des informations d'état... Fait
Paquets suggérés :
  libsasl2-modules-gssapi-mit | libsasl2-modules-gssapi-heimdal
Les NOUVEAUX paquets suivants seront installés :
```

on nous demande de choisir un mot de passe pour l'administrateur



on retape le mot de passe



## Paramètres principaux du fichier de configuration :

Ici nous utiliserons la méthode de config dans un fichier . Pour ça il faut :

-**Imp**érativement renommer le répertoire `/etc/ldap/slapd.d/` en `/etc/ldap/slapd.d.old/` comme

ceci :

```
root@asterisk-ldap:~# mv /etc/ldap/slapd.d/ /etc/ldap/slapd.d.old/
root@asterisk-ldap:~#
```

**\*\*Et\*\*** copier un fichier d'exemple de configuration d'un serveur LDAP qui se trouve dans

`/usr/share/slapd/slapd.conf` comme ceci :

```
root@asterisk-ldap:~# cp /usr/share/slapd/slapd.conf /etc/ldap/
root@asterisk-ldap:~#
```

```
root@gasterisk-ldap:/etc/ldap# ls
ldap.conf  sasl2  schema  slapd.conf  slapd.d.old
```

Maintenant éditons le fichier **slapd.conf** copier dans **/etc/ldap/** et y mettre la configuration qui se trouve dans le tableau suivant :

Nom paramètre	Signification	Valeur possible
moduleload	Pilote de base de donnée à charger	back_hdb
backend	Le backend est en fait le « moteur » permettant le stockage ou la récupération de donnée en réponse à une requête LDAP dans un annuaire	hdb
database	Type de base de donnée	hdb
suffixe	Le nom de la racine de l'annuaire	'dc=smarttech,dc=sn '
rootdn	Le dn de l'administrateur	'cn=admin,dc=smarttech,dc=sn '
rootpw(il faut ajouter en bas de rootdn)	Mot de passe de l'administrateur	passer
access	Liste d'accès à une entrée	o hange attrs=userPassword,shadowLastC by dn='cn=admin,dc=smarttech,dc=sn ' write by anonymous by self write by * none
access		to  by dn='cn=admin,dc=smarttech,dc=sn ' , write  by read

voici quelques exemples:

```
GNU nano 2.9.3                                slapd.conf                                Modifié

loglevel          none

# Where the dynamically loaded modules are stored
```

```

modulepath      /usr/lib/ldap
moduleload      back_hdb

# The maximum number of entries that is returned for a search operation
sizelimit 500

# The tool-threads parameter sets the actual amount of cpu's that is used
# for indexing.
tool-threads 1

#####
# Specific Backend Directives for @BACKEND@:
# Backend specific directives apply to this backend until another
# 'backend' directive occurs
backend         hdb

```

```

# 'database' directive occurs
database        hdb

# The base of your directory in database #1
suffix          "dc=smarttech,dc=sn"

# rootdn directive for specifying a superuser on the database. This is needed
# for syncrepl.
rootdn          "cn=admin,dc=smarttech,dc=sn"
rootpw          passer

```

```

# These access lines apply to database #1 only
access to attrs=userPassword,shadowLastChange
  by dn="cn=admin,dc=smarttech,dc=sn" write
  by anonymous auth
  by self write
  by * none

```

```

GNU nano 2.9.3                                slapd.conf                                Modifié

# want SASL (and possible other things) to work
# happily.
access to dn.base="" by * read

# The admin dn has full write access, everyone else
# can read everything.
access to *
  by dn="cn=admin,dc=smarttech,dc=sn" write
  by * read

```

Enregistrons et redémarrer le serveur comme suit :

```

root@asterisk-ldap:/etc/ldap# service slapd start
root@asterisk-ldap:/etc/ldap#

```

Vérifions si le serveur écoute sur le port **\*\*389\*\*** comme suit :

```

root@asterisk-ldap:/etc/ldap# netstat -anp | grep -w 389
tcp        0      0 0.0.0.0:389          0.0.0.0:*           LISTEN      9258/s1

```

```
tcp6      0      0 :::389      :::*        LISTEN    9258/s
root@asterisk-ldap:/etc/ldap#
```

Nous constatons que le serveur est t'en Bon état et que l'algorithme a été claire ....

Pour le teste nous allons paramétrer le client ldap...

comme je suis en local j'utilise la même machine comme serveur et client ...

Le fichier de configuration du client est dans `/etc/ldap/ldap.conf` nous allons lui renseigner deux

(2) paramètres qui sont :

- Le nom de la racine de l'annuaire
- L'adresse Ip du serveur LDAP

```
GNU nano 2.9.3      ldap.conf      Modifié

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

BASE    dc=smarttech,dc=sn
URI      ldap://192.168.158.128

#SIZELIMIT 12
#TIMELIMIT 15
#DEREF never

# TLS certificates (needed for GnuTLS)
TLS_CACERT    /etc/ssl/certs/ca-certificates.crt
```

Test de la connexion client-serveur : Pour tester si la connexion entre le client et le serveur fonctionne c'est simple

Il faut ouvrir une console chez le client et tapez la commande suivante :

```
root@asterisk-ldap:/etc/ldap# ldapsearch -x
# extended LDIF
#
# LDAPv3
# base <dc=smarttech,dc=sn> (default) with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# search result
search: 2
result: 32 No such object

# numResponses: 1
root@asterisk-ldap:/etc/ldap#
```

## II- Couplage Asterisk et LDAP

---

Avant toute chose, nous allons indiquer à notre \*\*LDAP\*\* les variables à prendre en compte pour

**Asterisk**. Ceci se passe dans un schéma disponible Cliquez-ici. Copier l'ensemble de ce fichier dans

un fichier [asterisk.schema](#) que vous placerez dans le répertoire [/etc/ldap/schema](#).

Il faut ensuite indiquer à LDAP de prendre en compte ce schéma. Pour cela, il faut ajouter ceci dans

le fichier de configuration [/etc/ldap/slapd.conf](#) comme ceci :

```
GNU nano 2.9.3                                slapd.conf                                Modifié

# This is the main slapd configuration file. See slapd.conf(5) for more
# info on the configuration options.

#####
# Global Directives:

# Schema and objectClass definitions
include      /etc/ldap/schema/core.schema
include      /etc/ldap/schema/cosine.schema
include      /etc/ldap/schema/nis.schema
include      /etc/ldap/schema/inetorgperson.schema
include      /etc/ldap/schema/asterisk.schema

# Where the pid file is put. The init.d script
# will not stop the server if you change this.
pidfile      /var/run/slapd/slapd.pid
```

Pour que cette modification soit prises en compte, il faut relancer le serveur LDAP :

```
root@asterisk-ldap:/etc/ldap# /etc/init.d/asterisk start
[ ok ] Starting slapd (via systemctl): slapd.service.
root@asterisk-ldap:/etc/ldap#
```

### Configuration de l'annuaire LDAP

---

Afin de stocker les paramètres des comptes SIP de l'Asterisk, il faut que nous ayons un endroit où les stockés.

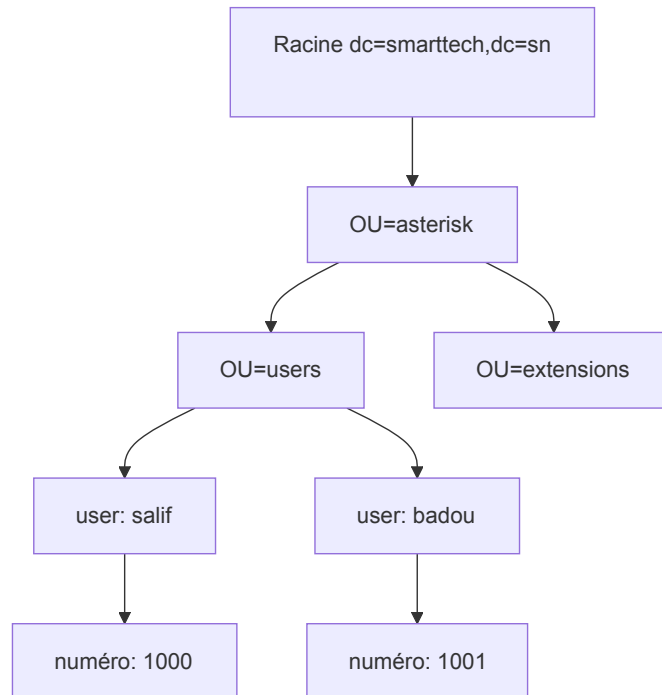
J'ai décidé de séparer ces deux informations par soucis de lisibilité. J'ai donc

choisi de créer une OU (Organizational Unit) dédiée à Asterisk. J'ai donc créer une OU Asterisk et

avec deux "sous-OU" users et extensions. Vous pouvez ajouter cette configuration comme cela :

Tout d'abord nous allons créer un fichier **LDIF** nommé **racine.ldif** dans **/etc/ldap/** contenant la

racine de notre annuaire LDAP comme ceci :



```
GNU nano 2.9.3                               racine.ldif
# racine.ldif
dn: dc=smarttech,dc=sn
objectClass: dcObject
objectClass: organization
dc: smarttech
o: smarttech.sn
```

On alimente avec la commande suivante

```
root@asterisk-ldap:/etc/ldap# ldapadd -x -D "cn=admin,dc=smarttech,dc=sn" -W -f racine.ldif

Enter LDAP Password:
adding new entry "dc=smarttech,dc=sn"

root@asterisk-ldap:/etc/ldap#
```

Nous allons afficher les entrées de l'annuaire comme ceci:

```
root@asterisk-ldap:/etc/ldap# ldapsearch -x
# extended LDIF
#
# LDAPv3
# base <dc=smarttech,dc=sn> (default) with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# smarttech.sn
dn: dc=smarttech,dc=sn
objectClass: organization
objectClass: dcObject
o: smarttech
dc: smarttech

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

```
# numResponses: 2
# numEntries: 1
root@asterisk-ldap:/etc/ldap#
```

Ensuite nous allons créer aussi un fichier **LDIF** nommé **info.ldif** dans **/etc/ldap/** contenant la information d'asterisk et celle de l'annuaire comme ceci :

```
GNU nano 2.9.3 info.ldif
# OU asterisk
dn: ou=asterisk,dc=smarttech,dc=sn
objectClass: top
objectClass: organizationalUnit
ou: asterisk

# OU users
dn: ou=users,ou=asterisk,dc=smarttech,dc=sn
objectClass: top
objectClass: organizationalUnit
ou: users

# OU extensions
dn: ou=extensions,ou=asterisk,dc=smarttech,dc=sn
objectClass: top
objectClass: organizationalUnit
ou: extensions
```

Il faut ensuite ajouter ce fichier **LDIF** à notre arborescence **LDAP**, pour cela il faut utiliser la commande suivante comme suit dans un terminal :

```
root@asterisk-ldap:/etc/ldap# ldapadd -x -D "cn=admin,dc=smarttech,dc=sn" -W -f info.ldif

Enter LDAP Password:
adding new entry "ou=asterisk,dc=smarttech,dc=sn"
adding new entry "ou=users,ou=asterisk,dc=smarttech,dc=sn"
adding new entry "ou=extensions,ou=asterisk,dc=smarttech,dc=sn"
```

Nous allons afficher les entrées de l'annuaire encore:

```
# OU asterisk
dn: ou=asterisk,dc=smarttech,dc=sn
objectClass: top
objectClass: organizationalUnit
ou: asterisk

# OU users
dn: ou=users,ou=asterisk,dc=smarttech,dc=sn
objectClass: top
objectClass: organizationalUnit
ou: users

# OU extensions
dn: ou=extensions,ou=asterisk,dc=smarttech,dc=sn
objectClass: top
objectClass: organizationalUnit
ou: extensions

# search result
search: 2
```



```
result: 0 Success
```

## II-a Connexion au serveur LDAP

Configuration de la connexion dans /etc/asterisk/res\_ldap.conf :

```
[_general]
host=192.168.158.128
port=389
protocol=3
basedn=ou=asterisk,dc=smarttech,dc=sn
pass=passer
user=cn=admin,dc=smarttech,dc=sn

[sip]
name = uid
callerid = AstAccountCallerID
canreinvite = AstAccountCanReinvite
context = AstAccountContext
host = AstAccountHost
type = AstAccountType
mailbox = AstAccountMailbox
md5secret = AstAccountRealmedPassword
fullcontact = AstAccountFullContact
nat = AstAccountNAT
qualify = AstAccountQualify
allow = AstAccountAllowedCodec
useragent = AstAccountUserAgent
lastms = AstAccountLastQualifyMilliseconds
additionalFilter=(objectClass=AsteriskSIPUser)

[extensions]
context = AstContext
exten = AstExtension
priority = AstPriority
app = AstApplication
appdata = AstApplicationData
additionalFilter=(objectClass=AsteriskExtension)
```

```
GNU nano 2.9.3          res_ldap.conf          Modifié

; Note that you can configure an ldaps: url here to get TLS support.
; Detailed configuration of certificates and supported CAs is done in your
; ldap.conf file for OpenLDAP clients on your system.
; This requires that you have OpenLDAP libraries compiled with TLS support

; *****
; NOTE: res_ldap.conf should be chmod 600 because it contains the plain-text LDAP
; password to an account with WRITE access to the asterisk configuration.
; *****

[_general]
;
; Specify one of either host and port OR url. URL is preferred, as you can
; use more options.
host=192.168.158.128          ; LDAP host
port=389
;url=ldap://ldap3.mydomain.com:3890
protocol=3                   ; Version of the LDAP protocol to use; default is 3
basedn=ou=asterisk,dc=smarttech,dc=sn ; Base DN
user=cn=admin,dc=smarttech,dc=sn      ; Bind DN
pass=passer                   ; Bind password
```

```
[extensions]
context = AstExtensionContext
exten = AstExtensionExten
priority = AstExtensionPriority
app = AstExtensionApplication
appdata = AstExtensionApplicationData
additionalFilter=(objectClass=AstExtension)
```

```
[sip]
name = uid ; We use the "cn" as the default value for name
; because objectClass=AsteriskSIPUser does not allow
; If your entry combines other objectClasses and you
; prefer to change the line to be name = uid, es
; contain spaces in the cn field.
; You may also find it appropriate to use someth
; This is possible by changing the line above to
; prefer).
;

amaflags = AstAccountAMAFlags
callgroup = AstAccountCallGroup
callerid = AstAccountCallerID
canreinvite = AstAccountCanReinvite
directmedia = AstAccountDirectMedia
context = AstAccountContext
dtmfmode = AstAccountDTMFMode
fromuser = AstAccountFromUser
fromdomain = AstAccountFromDomain
fullcontact = AstAccountFullContact
```

```
fullcontact = gecoc
host = AstAccountHost
insecure = AstAccountInsecure
mailbox = AstAccountMailbox
md5secret = AstAccountRealmedPassword ; Must be an MD5 hash
; {md5} but it
; Generate the hash
; echo "my_pass"

nat = AstAccountNAT
deny = AstAccountDeny
permit = AstAccountPermit
pickupgroup = AstAccountPickupGroup
port = AstAccountPort
qualify = AstAccountQualify
restrictcid = AstAccountRestrictCID
rtptimeout = AstAccountRTPTIMEOUT
```

```
type = AstAccountType
useragent = AstAccountUserAgent
disallow = AstAccountDisallowedCodec
allow = AstAccountAllowedCodec
MusicOnHold = AstAccountMusicOnHold
regseconds = AstAccountExpirationTimestamp
regcontext = AstAccountRegistrationContext
```

```
regexen = AstAccountRegistrationExten
CanCallForward = AstAccountCanCallForward
ipaddr = AstAccountIPAddress
defaultuser = AstAccountDefaultUser
regserver = AstAccountRegistrationServer
lastms = AstAccountLastQualifyMilliseconds
supportpath = AstAccountPathSupport
additionalFilter=(objectClass=AsteriskSIPUser)
```

Comme vous pouvez le voir la section `**[sip]**` permet de faire la translation entre les variables

Asterisk et les variables LDAP afin que les deux serveur puissent se "comprendre".

Il en est de même pour la partie `[extensions]`.

Une fois que nous avons défini la connexion entre le serveur [Asterisk](#) et le serveur LDAP, il faut

dire à [Asterisk](#) où il doit aller chercher les paramètres des utilisateurs SIP. Pour cela, il faut

modifier le fichier `/etc/asterisk/extconfig.conf` comme ceci :

```
; The only option available currently is the 'p' option, which disallows
; extension pattern queries to the database. If you have no patterns defined
; in a particular context, this will save quite a bit of CPU time. However,
; note that using dynamic realtime extensions is not recommended anymore as a
; best practice; instead, you should consider writing a static dialplan with
; proper data abstraction via a tool like func_odbc.

sipusers => ldap,"ou=users,ou=asterisk,dc=barry,dc=sn",sip
sippeers => ldap,"ou=users,ou=asterisk,dc=barry,dc=sn",sip
extensions => ldap,"ou=extensions,ou=asterisk,dc=barry,dc=sn",extensions
```

## II-b Configuration générale des comptes SIP

---

Configuration dans `/etc/asterisk/sip.conf` :

```
[general]
rtcachefriends=yes
callevnts=yes
realm=smarttech.sn
```

L'authentification des utilisateur SIP. Si vous utilisez déjà un Asterisk, veuillez rajouter ces

Paramètres sans écraser les anciens contenus dans votre fichier.

- ❑ `rtcachefriends=yes` //permet de mettre en cache les infos des utilisateurs (obligatoire car elle permet de garder en mémoire l'adresse IP avec laquelle l'utilisateur s'est connecté. Sans ce paramètre l'appel n'aboutira pas car l'Asterisk ne saura pas trouver les utilisateurs.
- ❑ `callevnts=yes` //permet de remonter les informations concernant un appel
- ❑ `realm=smarttech.sn` //nom de domaine géré par l'annuaire LDAP

## II-c Configuration générale du dialplan

---

Configuration dans /etc/asterisk/extensions.conf :

```
GNU nano 2.9.3                                extensions.conf

exten => _X.,n,Wait(1.25)
exten => _X.,n,SayDigits(${CALLERID(ani)})      ; playback again in case of mis$
exten => _X.,n,Return()

; For more information on applications, just type "core show applications" at y$
; friendly Asterisk CLI prompt.
;
; "core show application <command>" will show details of how you
; use that particular application in this file, the dial plan.
; "core show functions" will list all dialplan functions
; "core show function <COMMAND>" will show you more information about
; one function. Remember that function names are UPPER CASE.

[internal]
switch => Realtime/@"
```

## Vérification de la connexion au LDAP

Avant toute chose, il faut recharger la configuration afin que les modifications que l'on a apporté

soient prises en compte. Pour cela, il faut se connecter au CLI d'Asterisk :

```
root@asterisk-ldap:/etc/asterisk# asterisk -vvvvvvvvvvvvvvvvvvvvv  
  
Asterisk 17.6.0, Copyright (C) 1999 - 2018, Digium, Inc. and others.  
Created by Mark Spencer <markster@digium.com>  
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.  
This is free software, with components licensed under the GNU General Public License version  
=====
```

Connected to Asterisk 17.6.0 currently running on bhpro (pid = 1258)

```
asterisk-ldap*CLI> reload  
-- Reloading module 'extconfig' (Configuration)
```

Mettre plusieurs **\*\*v\*\*** dans la commande permet d'augmenter le niveau de debug. Il faut ensuite taper les commandes suivantes dans le CLI:

```
asterisk-ldap*CLI> module reload
```

cette commande permet de recharger tous les modules qu'asterisk a charger notamment le module LDAP puis chargeons le fichier pjsip.conf

```
asterisk-ldap*CLI> pjsip reload
```

Nous rechargeons ensuite le fichier extensions.conf

```
asterisk-ldap*CLI> dialplan reload
Dialplan reloaded.

== Setting global variable 'CONSOLE' to 'Console/dsp'
== Setting global variable 'IAXINFO' to 'guest'
== Setting global variable 'TRUNK' to 'DAHDI/G2'
== Setting global variable 'TRUNKMSD' to '1'
-- Including switch 'DUNDi/e164' in context 'dundi-e164-switch'
-- Including switch 'Realtime/q' in context 'internal'
-- Including switch 'Lua/' in context 'public'
-- Including switch 'Lua/' in context 'demo'
-- Including switch 'Lua/' in context 'local'
-- Including switch 'Lua/' in context 'default'
-- Including switch 'DUNDi/e164' in context 'ael-dundi-e164-switch'
-- Time to scan old dialplan and merge leftovers back into the new: 0.000605 sec
-- Time to restore hints and swap in new dialplan: 0.000016 sec
-- Time to delete the old dialplan: 0.000537 sec
-- Total time merge_contexts_delete: 0.001158 sec
-- pbx_config successfully loaded 51 contexts (enable debug for details).

asterisk-ldap*CLI>
```

La commande **realtime show ldap status** permet de connaître l'état de la connexion entre **le**

serveur Asterisk et le serveur LDAP.

```
asterisk-ldap*CLI> realtime show ldap status
Connected to 'ldap://192.168.158.1278:389', baseDN ou=asterisk,dc=barry,dc=sn with username c
asterisk-ldap*CLI>
```



## Ajout des utilisateurs

Maintenant que nos deux serveurs communiquent correctement ensemble, nous pouvons ajouter les

utilisateurs dans notre annuaire LDAP. Pour ça on a un script python pour ajouter des users ldap une version via la console et une autre avec tkinter tous les deux doivent être accompagné d'un troisieme fichier bash adduser.sh Les voici : users-gui.py , users.py addusers.sh

## Test de la configuration

voici une

video montrant comment executer le script les resultat et on va utiliser un outil pour appeler zoiper

Création de deux utilisateurs pour le test :

- leuk (extension: 3000) - bouki(extension: 3001)

