

LDAP

- Création Docker OPENLDAP image OSIXIA / GITHUB avec réplication
- Création OpenLdap
- Passage Idif, changement des information
- OpenLdap commande

Création Docker OPENLDAP image OSIXIA / GITHUB avec réplication

```
#####
creation fake Docker sur le serveur ldapPRéPRD-02:
LDAP_CID=$(docker run --restart always --name ldap-test --hostname "hostname" --env LDAP_BASE_DN="DC" --env LDAP_ORGANISATION="NOM" --env LDAP_DOMAIN="domaine" --env LDAP_ADMIN_PASSWORD=PASSWORD --env LDAP_TLS=false -v /data2/ldap:/var/lib/ldap -v /data2/slapd.d:/etc/ldap/slapd.d -p 1389:389 --detach osixia/openldap:1.5.0)

#####
creation Vrai Docker sur srv 1 :
LDAP_CID=$(docker run --restart always --name ldap --hostname "hostname" --env LDAP_BASE_DN="DC" --env LDAP_ORGANISATION="NOM" --env LDAP_DOMAIN="domaine" --env LDAP_ADMIN_PASSWORD=PASSWORD --env LDAP_TLS=true --env LDAP_TLS_VERIFY_CLIENT=never --env LDAP_REPLICATION=true --env LDAP_REPLICATION_HOSTS="#PYTHON2BASH:['ldap://hostname', 'ldap://hostname']" --env LDAP_TLS_CERT_FILE=/etc/pki/openldap/pprod-2/container/service/slapd/assets/certs -p 389:389 -p 636:636 --detach osixia/openldap:1.5.0)

creation Vrai Docker sur srv 2 :
LDAP_CID=$(docker run --restart always --name ldap --hostname "hostname" --env LDAP_BASE_DN="DC" --env LDAP_ORGANISATION="NOM" --env LDAP_DOMAIN="domaine" --env LDAP_ADMIN_PASSWORD=PASSWORD --env LDAP_TLS=true --env LDAP_TLS_VERIFY_CLIENT=never --env LDAP_REPLICATION=true --env LDAP_REPLICATION_HOSTS="#PYTHON2BASH:['ldap://hostname', 'ldap://hostname']" --env LDAP_TLS_CERT_FILE=/etc/pki/openldap/pprod-2/container/service/slapd/assets/certs -p 389:389 -p 636:636 --detach osixia/openldap:1.5.0)
```

creation fake Docker sur le serveur : CREATION LDAP TEST SANS REPLICATION POUR TESTER un LDIF.

```
LDAP_CID=$(docker run --restart always --name ldap-test --hostname "hostname" --env LDAP_BASE_DN="DC" --env LDAP_ORGANISATION="NOM" --env LDAP_DOMAIN="domaine" --env LDAP_ADMIN_PASSWORD=PASSWORD --env LDAP_TLS=false -v /data2/ldap:/var/lib/ldap -v /data2/slapd.d:/etc/ldap/slapd.d -p 1389:389 --detach osixia/openldap:1.5.0)
```

```
#####
#####
#####
```

creation Vrai Docker sur srv 1 avec replication et sécurité certificat ect.. = c'est une commande entiere a cp

```
LDAP_CID=$(docker run --restart always --name ldap --hostname "hostname" --env LDAP_BASE_DN="DC" --env LDAP_ORGANISATION="NOM" --env LDAP_DOMAIN="domaine" --env LDAP_ADMIN_PASSWORD=PASSWORD --env LDAP_TLS=true --env LDAP_TLS_VERIFY_CLIENT=never --env LDAP_REPLICATION=true --env
```

```
LDAP_REPLICATION_HOSTS="#PYTHON2BASH:['ldap://noomdulap','ldap://noomdulap']" --env
LDAP_TLS_CERT_FILENAME=cert.crt --env LDAP_TLS_KEY_FILENAME=cert.key --env
LDAP_TLS_CA_CERT_FILENAME=ca.crt -v /data/ldap:/var/lib/ldap -v /data/slapd.d:/etc/ldap/slapd.d --volume
/etc/pki/openldap/pprod-2:/container/service/slapd/assets/certs -p 389:389 -p 636:636 --detach
osixia/openldap:1.5.0)
```

creation Vrai Docker sur srv 2 avec replication sur le 1 et sécurité certificat ect.. = c'est une commande entiere a cp

```
LDAP_CID=$(docker run --restart always --name ldap --hostname "hostname" --env LDAP_BASE_DN="DC" --env
LDAP_ORGANISATION="NOM" --env LDAP_DOMAIN="domaine" --env LDAP_ADMIN_PASSWORD=PASSWORD --
env LDAP_TLS=true --env LDAP_TLS_VERIFY_CLIENT=never --env LDAP_REPLICATION=true --env
LDAP_REPLICATION_HOSTS="#PYTHON2BASH:['ldap://noomdulap','ldap://noomdulap']" --env
LDAP_TLS_CERT_FILENAME=cert.crt --env LDAP_TLS_KEY_FILENAME=cert.key --env
LDAP_TLS_CA_CERT_FILENAME=ca.crt -v /data/ldap:/var/lib/ldap -v /data/slapd.d:/etc/ldap/slapd.d --volume
/etc/pki/openldap/pprod-2:/container/service/slapd/assets/certs -p 389:389 -p 636:636 --detach
osixia/openldap:1.5.0)
```

Suppression du docker et des data docker, vite fait et efficace, vous pouvez après recréer le ldap dockeriser avec les commande du dessus :

```
##### suppression data + dockre
docker stop ""
docker rm ldap
rm -rf /data2/slapd.d/*
rm -rf /data2/ldap/*
```

```
#####
#####
#####
```

Outils en interface sur windows qui aide a rentrer des Idif ou troublehsoot des compte ect ...

```
vi LDAP/LdapAdminTool-7.8.x-win-x64-Setup.exe
```

commande test ldap et autres voir si il répond.

```
ldapsearch -x -H ldap://noomdulap -b dc=,dc= -D "cn=,dc=,dc=" -W
ldappasswd -D "cn=,dc=,dc=" -W -S "uid=mail@domaine.com,ou=,ou=,dc=,dc="
ldapwhoami -x -D "uid=mail@domaine.com,ou=,ou=,dc=,dc=" -W -v
```

```
ldapsearch -x -H ldap://noomduldap -b dc=,dc= -D "cn=,dc=,dc=" -W
ldapwhoami -x -D "uid=mail@domaine.com,ou=,ou=,dc=,dc=" -W -v
ldapwhoami -x -D "uid=,ou=,ou=,dc=,dc=" -W -v
```

grep -5ni 'test.test,' LdapExt-Prod-FINAL.ldif == pour rechercher les user ou erreur dans un contexte chaîne de caractère sur du dn ou uid

commande test ldap et autres voir si il répond.

```
[admin] $ docker exec -it ldap /bin/bash

***** à l'interieur du docker *****

ldappasswd -D "cn=admin" -W -S "uid=,ou=,dc=,dc="
ldapwhoami -x -D "uid=,com,ou=,ou=,dc=,dc=" -W -v

***** another server searching docker ldap *****

ldapsearch -x -H ldap:// -b dc=,dc= -D "cn=admin,dc=,dc=" -W
ldapsearch -x -H ldap://10.105 -b dc=,dc= -D "cn=admin,dc=,dc=" -W


***** docker cmd ! *****
docker exec -it ldap /bin/bash

docker exec ldap ldapadd -x -D "cn=admin,dc=,dc=" -W -f /home/.ldif
docker cp ./ .ldif ldap:/home/.ldif
docker logs ldap
```

Debug docker logs :

3. Suivre les logs en temps réel en affichant d'abord les 50 dernières lignes :

```
bash
```

 Copier le code

```
docker logs --tail 50 -f 123abc456def
```

Créer un ldif :

```
ldapsearch -x -H ldap://localhost -D "cn=admin,dc=example,dc=com" -w admin_password -b
"dc=example,dc=com" "(objectClass=*)" > backup.ldif
```

Création OpenLdap

Install package openldap, on utilise yum pour éviter les problèmes de dépendances

```
yum -y install openldap* migrationtools
```

```
[root@██████████ ~]# yum -y install openldap* migrationtools
```

Une fois installé, créé le Password admin du ldap

```
slappaswd
```

```
[root@██████████ ~]# slappaswd  
New password:  
Re-enter new password:  
{SSHA}nlrrt████████████████████
```

Copier le SHA pour l'intégrer plus tard dans le fichier de configuration

Accéder au dossier config du ldap et modifier le fichier olcDatabase={2}hdb.ldif

```
[root@██████████ ~]# cd /etc/openldap/slapd.d/cn=config  
[root@██████████ cn=config]# vi olcDatabase={2}hdb.ldif
```

Modifier les paramètres "olcSuffix" et "olcRootDN" selon votre domaine et ajouter "olcRootPW"

```
# AUTO-GENERATED FILE - DO NOT EDIT!! Use ldapmodify.
# CRC32 9f295b19
dn: olcDatabase={2}hdb
objectClass: olcDatabaseConfig
objectClass: olcHdbConfig
olcDatabase: {2}hdb
olcDbDirectory: /var/lib/ldap
olcSuffix: dc=[REDACTED],dc=fr
olcRootDN: cn=Manager,dc=[REDACTED],dc=fr
olcDbIndex: objectClass eq,pres
olcDbIndex: ou,cn,mail,surname,givenname eq,pres,sub
olcRootPW: [REDACTED]
structuralObjectClass: olcHdbConfig
entryUUID: 02d0ac38-2d37-103e-822b-fb904797972b
creatorsName: cn=config
createTimestamp: 20231212123752Z
"olcDatabase={2}hdb.ldif" 19L, 657C
```

Ajout des droits de monitor au compte en modifiant le fichier olcDatabase={1}monitor.ldif :

```
[root@[REDACTED] cn=config]# vi olcDatabase={1}monitor.ldif
```

Modifier les paramètres "olcAccess" de votre domaine et ajouter

```
# AUTO-GENERATED FILE - DO NOT EDIT!! Use ldapmodify.
# CRC32 bd9e375b
dn: olcDatabase={1}monitor
objectClass: olcDatabaseConfig
olcDatabase: {1}monitor
olcAccess: {0}to * by dn.base="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth" read by dn.base="cn=Manager,[REDACTED],dc=fr" read by * none
structuralObjectClass: olcDatabaseConfig
entryUUID: 02d0a8b4-2d37-103e-822a-fb904797972b
creatorsName: cn=config
createTimestamp: 20231212123752Z
entryCSN: 20231212123752.474096Z#000000#000#000000
modifiersName: cn=config
modifyTimestamp: 20231212123752Z
~
~
"olcDatabase={1}monitor.ldif" 14L, 561C
```

Vérifier la configuration avec la commande suivante

```
slaptest -u
```

```
[root@[REDACTED] cn=config]# slaptest -u
65785521 ldif_read_file: checksum error on "/etc/openldap/slapd.d/cn=config/olcDatabase={1}monitor.ldif"
65785521 ldif_read_file: checksum error on "/etc/openldap/slapd.d/cn=config/olcDatabase={2}hdb.ldif"
config file testing succeeded
```

Nous pouvons ignorer les erreurs checksum

Démarrer le service, l'ajouter au démarrage automatique et vérifier le démarrage

```
[root@redhat7 ~]# systemctl start slapd
[root@redhat7 ~]# systemctl enable slapd
Created symlink from /etc/systemd/system/multi-user.target.wants/slapd.service to /usr/lib/systemd/system/slapd.service.
[root@redhat7 ~]# netstat -lt | grep ldap
bash: netstat: command not found...
[root@redhat7 ~]# netstat -lt | grep ldap
tcp        0      0 0.0.0.0:ldap 0.0.0.0:*      LISTEN
tcp6       0      0 :::ldap    :::*           LISTEN
```

Copier le sample de la base et attribuer les droits :

```
[root@redhat7 ~]# cp /usr/share/openldap-servers/DB_CONFIG.example /var/lib/ldap/DB_CONFIG
[root@redhat7 ~]# chown -R ldap:ldap /var/lib/ldap
```

Crée les schéma pour le ldap :

```
[root@redhat7 ~]# ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/cosine.ldif
[root@redhat7 ~]# ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/nis.ldif
[root@redhat7 ~]# ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/inetorgperson.ldif
```

Un annuaire d'un site applicatif : pour répondre sur le ldap

```
4 <!-- Annuaire OpenLDAP -->
5 <add key="Ldap_ServerName" value="OpenLDAP interne" />
6 <add key="Ldap_Type" value="Open" />
7 <add key="Ldap_Serveur" value="17.105:389" />
8 <add key="Ldap_OrganisationUtilisateurs" value="ou=" />
9 <add key="Ldap_Partition" value="dc=" />
10 <add key="Ldap_Code" value="uid" />
11 <add key="Ldap_Email" value="mail" />
12 <add key="Ldap_Nom" value="sn" />
13 <add key="Ldap_Prenom" value="givenName" />
14 <add key="Ldap_ServeurSsl" value="" />
15 <add key="Ldap_Utilisateur" value="" />
16 <add key="Ldap_MotDePasse" value="" />
17 <add key="AuthenticationCreationUtilisateur" value="False" />
18 <!-- Libellé du profil à associer par défaut aux utilisateurs créés automatiquement (clé : Client-Environnement) -->
19 <add key="AuthenticationProfilDefaut,Prod" value="" />
20
21 <add key="Ldap_EnablePasswordChange" value="True" />
22 <add key="Ldap_EnableLdapUserUpdate" value="True" />
23 <add key="Ldap_EnableLdapUserCreate" value="True" />
24 <add key="Ldap_EnableLdapUserDelete" value="True" />
25 <add key="Ldap_SecurityErrorMessage" value="Votre compte sera désactivé pendant 5 minutes après 3 échecs d'authentification." />
26 <add key="Ldap_Member" value="" />
27 <add key="Ldap_SecurityGroupDnToCheck" value="" />
28
29 <!-- Blocage accès Back -->
30 <add key="LoginPage_DisableLdap" value="True" />
31
32
33 </appSettings>
```

Passage Idif, changement des information

Passage d'un Idif qui parvient d'une autre entité, il faut passer ce script pour retirer les informations du fichier Idif pour ne pas avoir l'erreur **CONTRAINTE 61**

```
#SCRIPT
#!/bin/bash
echo "inupt file LdapExt-Prod.Idif "cat $1 | grep --binary-files=text -vi structuralObjectClass > tmp1.Idif
cat tmp1.Idif | grep --binary-files=text -vi entryUUID > tmp2.Idif
cat tmp2.Idif | grep --binary-files=text -vi creatorsName > tmp3.Idif
cat tmp3.Idif | grep --binary-files=text -vi createTimestamp > tmp4.Idif
cat tmp4.Idif | grep --binary-files=text -vi entryCSN > tmp5.Idif
cat tmp5.Idif | grep --binary-files=text -vi modifiersName > tmp6.Idif
cat tmp6.Idif | grep --binary-files=text -vi modifiersName > tmp7.Idif
cat tmp7.Idif | grep --binary-files=text -vi modifyTimestamp > tmp8.Idif
cat tmp8.Idif | grep --binary-files=text -vi contextCSN > $1-final.Idif
echo "conversion en cours ...\n...\n...."
echo " fin Conversion , OutPUT file : import-v2.Idif "
# + suppression de l'ou et de l'admin
```

Il faut : `echo " fin Conversion , OutPUT file : import-v2.Idif "`
`# + suppression de l'ou et de l'admin`
`~`
`~`
script .

OpenLdap commande

1. `Idapwhoami -x -D "uid=*** -W -v`
 - Cette commande permet de s'authentifier avec l'utilisateur spécifié et affiche l'identité LDAP de l'utilisateur actuel (fonction `whoami`).
2. `Idapsearch -x -LLL -D "cn=admin,dc=***,dc=***" -W -b "ou=extern,ou=users,dc=***,dc=***" "(uid=*)" | grep dn`
 - Effectue une recherche dans l'annuaire LDAP en filtrant sur l'attribut `uid` et extrait les entrées de type `dn` (distinguished name).
3. `cat comptes_geode_extern.ldif | grep -5ni granet`
 - Affiche les 5 lignes avant et après chaque occurrence de "granet" dans le fichier `comptes_geode_extern.ldif`.
4. `Idapadd -x -D "cn=admin,dc=***,dc=***" -W -f comptes_geode_extern.ldif`
 - Ajoute des entrées LDAP à partir du fichier `comptes_geode_extern.ldif`.
5. `Idapdelete -x -r "ou=users,dc=***,dc=***" -W -D "cn=admin,dc=***,dc=***"`
 - Supprime des entrées récursivement dans l'OU `users` de l'annuaire LDAP.
6. `Idapsearch -x -LLL -D "cn=admin,dc=***,dc=***" -W -b "cn=admin,dc=***,dc=***" "(objectClass=*)"`
 - Effectue une recherche LDAP sur l'OU admin et récupère toutes les entrées de ce conteneur.
7. `Idapwhoami -x -D "cn=admin-byzance,dc=***,dc=***" -W -H ldap://***`
 - S'authentifie sur un serveur LDAP distant et affiche l'identité LDAP de l'utilisateur.
8. `Idapsearch -x -H ldap://*** -D "cn=admin,dc=***,dc=***" -W -b "dc=***,dc=***" "(objectClass=*)" > backup-jimmy-test.ldif`
 - Effectue une recherche sur le serveur LDAP distant et exporte les résultats dans un fichier `backup-jimmy-test.ldif`.
9. `Idapsearch -x -H ldap://*** -D "cn=admin,dc=***,dc=***" -W -b "ou=vm,ou=users,dc=***,dc=***" "(objectClass=*)" > ou-vm.ldif`
 - Recherche les objets dans l'OU `vm` de l'annuaire LDAP distant et exporte les résultats dans le fichier `OU-VM.ldif`.
10. `Idapsearch -x -H ldap://*** -D "cn=admin,dc=***,dc=***" -W -b "ou=byzance,ou=users,dc=***,dc=***" "(objectClass=*)" | grep -c "^dn: "`
 - Recherche les objets dans l'OU `byzance` et compte le nombre d'entrées dans le résultat.

11. `grep -c "^dn: uid=" OU-FULL-BACKUP-PROD/full-backup-prod-09-09-2024.ldif > directement sur un file.ldif`
- Recherche dans un fichier LDIF et compte le nombre d'entrées de type `dn` pour l'attribut `uid`.