OpenLDAP sur Debian : installer et configurer un annuaire LDAP d'entreprise

OpenLDAP sur Debian fournit un annuaire LDAP open source pour centraliser l'authentification et la gestion des identités. Ce guide couvre l'installation de slapd, la structuration des OUs, la configuration TLS avec Let's Encrypt, phpLDAPadmin et l'intégration PAM Linux.

OpenLDAP sur Debian est la solution open source de référence pour déployer un annuaire LDAP centralisé sans licence propriétaire. OpenLDAP, dont la version stable actuelle est 2.6.x, implémente le protocole LDAP (Lightweight Directory Access Protocol) v3 défini dans la RFC 4511, et constitue l'annuaire d'identités le plus déployé dans les environnements Linux : selon LinuxFoundation, plus de 40 % des infrastructures Linux utilisent LDAP pour la gestion centralisée des utilisateurs. L'installation sur Debian 12 Bookworm se fait via le package `slapd`, le démon OpenLDAP, complété par `ldap-utils` pour les outils en ligne de commande. OpenLDAP permet de centraliser les comptes utilisateurs, les groupes, les politiques de mot de passe et les accès SSH dans un annuaire unique, évitant la duplication des comptes sur chaque serveur. Ce guide pratique vous accompagne de l'installation à une configuration de production sécurisée : structure organisationnelle (OUs), ajout d'utilisateurs et groupes via LDIF, sécurisation TLS avec Let's Encrypt, interface d'administration phpLDAPadmin, intégration PAM Linux pour l'authentification centralisée, et bonnes pratiques de sauvegarde.

À RETENIR

À retenir

OpenLDAP 2.6.x sur Debian 12 : package `slapd` + `ldap-utils` — la configuration se fait désormais via cn=config (OLC) et non plus via `slapd.conf`, qui est obsolète depuis OpenLDAP 2.4.

Structure DIT hiérarchique : l'arborescence LDAP (Distinguished Name) s'organise en dc= (domaine), ou= (unités organisationnelles), uid= (utilisateurs), cn= (groupes) — planifier la structure avant de créer les entrées.

TLS obligatoire en production : les communications LDAP en clair sur le port 389 transmettent les mots de passe en clair — forcer LDAPS (port 636) ou StartTLS en production, particulièrement pour l'authentification PAM.

phpLDAPadmin simplifie l'administration : l'interface web PHP permet de créer, modifier et supprimer des entrées LDAP sans maîtriser la syntaxe LDIF, mais ne remplace pas la ligne de commande pour les opérations en masse.

Sauvegarde via slapcat : exporter l'annuaire avec `slapcat -l backup.ldif` — cette commande s'exécute sans arrêter slapd et produit un fichier LDIF complet restaurable avec `slapadd`.

Qu'est-ce qu'OpenLDAP et pourquoi le déployer sur Debian ?

OpenLDAP est l'implémentation open source libre du protocole LDAP (Lightweight Directory Access Protocol), développée depuis 1998 par la OpenLDAP Foundation. LDAP est un protocole client-serveur optimisé pour les opérations de lecture intensive sur des annuaires d'entreprise — il est 10 à 100 fois plus rapide que MySQL ou PostgreSQL pour les requêtes de type "chercher

un utilisateur par son login". OpenLDAP est utilisé comme backend d'annuaire par de nombreuses solutions : Samba (partage de fichiers Windows/Linux), Postfix/Dovecot (messagerie), Apache (authentification), Nextcloud, et des dizaines d'applications métier.

Sur Debian 12 Bookworm, OpenLDAP 2.6.x est disponible dans les dépôts officiels. Debian est la distribution de référence pour OpenLDAP dans les environnements serveur grâce à la stabilité de ses packages, aux scripts de maintenance intégrés (`dpkg-reconfigure slapd`) et à la qualité de la documentation. Comparé à Active Directory, OpenLDAP est plus léger, entièrement open source, mais ne fournit pas nativement le protocole Kerberos ni les stratégies de groupe Windows — des solutions comme FreeIPA combinent OpenLDAP, Kerberos et DNS pour une alternative complète à Active Directory.

Installation de slapd et ldap-utils sur Debian 12

L'installation d'OpenLDAP sur Debian 12 démarre par l'installation du package `slapd` . L'assistant de configuration dpkg demande le mot de passe administrateur LDAP et configure automatiquement la base DN à partir du nom de domaine du système.

```
 # Mise à jour des paquets avant installation
apt update && apt upgrade -y

# Installation d'OpenLDAP (slapd) et des outils LDAP
apt install -y slapd ldap-utils

# Le script d'installation demande :
# - Mot de passe administrateur LDAP (à conserver précieusement)
# L'assistant détecte le hostname et configure dc=domaine,dc=local automatiquement

# Reconfigurer slapd si nécessaire (modifier le DN de base, le mot de passe, etc.)
dpkg-reconfigure slapd
# Options recommandées :
# - Omit OpenLDAP server configuration? Non
# - DNS domain name: domaine.local (adapte dc=domaine,dc=local)
# - Organization name: Mon Organisation
# - Admin password: [mot de passe fort]
# - Database backend: MDB (recommandé, plus performant que BDB/HDB)
# - Remove the database when slapd is purged? Non
# - Move old database? Oui

# Vérifier le statut du service slapd
systemctl status slapd

# Test de connectivité LDAP en local
ldapsearch -x -H ldap://localhost -b "" -s base "(objectclass=*)" namingContexts
```

Package	Fonction	Port	Configuration
slapd	Démon OpenLDAP (serveur)	389 (LDAP), 636 (LDAPS)	/etc/ldap/slapd.d/
ldap-utils	Outils CLI (ldapsearch, ldapadd, ldapmodify)	—	/etc/ldap/ldap.conf
phpldapadmin	Interface web d'administration	80/443 via Apache	/etc/phpldapadmin/config.php
libnss-ldap / sssd	Intégration NSS/PAM pour auth Linux	—	/etc/sssd/sssd.conf

Comment structurer l'annuaire LDAP (DIT) ?

Avant d'ajouter des entrées, planifier la structure DIT (Directory Information Tree) est essentiel. Une mauvaise structure initiale est difficile à corriger sans reconstruire l'annuaire. La convention recommandée pour les entreprises organise l'arborescence en unités organisationnelles fonctionnelles.

```
 # Structure DIT recommandée :
# dc=domaine,dc=local
# ├── ou=users          (tous les comptes utilisateurs)
# │   ├── ou=admins     (comptes administrateurs)
# │   └── ou=prestataires (comptes prestataires)
# ├── ou=groups         (groupes Unix/Linux)
# ├── ou=services       (comptes de service applicatifs)
# └── ou=hosts          (entrées hôtes si nécessaire)

# Fichier LDIF pour créer la structure de base
cat > /tmp/structure_base.ldif << 'EOF'
# Unité organisationnelle : utilisateurs
dn: ou=users,dc=domaine,dc=local
objectClass: organizationalUnit
ou: users
description: Comptes utilisateurs

# Unité organisationnelle : groupes
dn: ou=groups,dc=domaine,dc=local
objectClass: organizationalUnit
ou: groups
description: Groupes Unix

# Unité organisationnelle : services
dn: ou=services,dc=domaine,dc=local
objectClass: organizationalUnit
ou: services
description: Comptes de service applicatifs
EOF

# Injecter la structure dans l'annuaire
ldapadd -x -H ldap://localhost -D "cn=admin,dc=domaine,dc=local"     -W -f /tmp/structure_base.ld

# Vérifier la structure créée
ldapsearch -x -H ldap://localhost -D "cn=admin,dc=domaine,dc=local"     -W -b "dc=domaine,dc=loca
```

Ajout d'utilisateurs et de groupes via LDIF

Les entrées LDAP se créent via des fichiers LDIF (LDAP Data Interchange Format). Chaque entrée LDIF définit le DN de l'objet, ses classes d'objets et ses attributs. Pour les comptes utilisateurs Linux, l'objectClass `posixAccount` est requis pour l'intégration NSS/PAM.

```
 # Générer un hash de mot de passe pour LDAP (nécessite slappasswd)
slappasswd -h {SSHA}
# Entrer le mot de passe quand demandé — copier le hash {SSHA}... généré

# Fichier LDIF pour créer un utilisateur (adapté aux systèmes Linux via posixAccount)
cat > /tmp/user_alice.ldif << 'EOF'
dn: uid=alice,ou=users,dc=domaine,dc=local
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: alice
sn: Dupont
givenName: Alice
cn: Alice Dupont
displayName: Alice Dupont
uidNumber: 10001
gidNumber: 10000
userPassword: {SSHA}HASH_GENERE_SLAPPASSWD_ICI
gecos: Alice Dupont
loginShell: /bin/bash
homeDirectory: /home/alice
mail: alice@domaine.local
EOF

# Ajouter l'utilisateur
ldapadd -x -H ldap://localhost -D "cn=admin,dc=domaine,dc=local"     -W -f /tmp/user_alice.ldif

# Créer un groupe et y ajouter Alice
cat > /tmp/group_sysadmins.ldif << 'EOF'
dn: cn=sysadmins,ou=groups,dc=domaine,dc=local
objectClass: posixGroup
cn: sysadmins
gidNumber: 10000
memberUid: alice
memberUid: bob
description: Administrateurs système
EOF

ldapadd -x -H ldap://localhost -D "cn=admin,dc=domaine,dc=local"     -W -f /tmp/group_sysadmins.l

# Rechercher un utilisateur
ldapsearch -x -H ldap://localhost -D "cn=admin,dc=domaine,dc=local"     -W -b "ou=users,dc=domain
```

Comment sécuriser OpenLDAP avec TLS (Let's Encrypt) ?

Les communications LDAP sans TLS transmettent les identifiants en clair sur le réseau — critique lors de l'authentification PAM Linux. Sur Debian 12, la configuration TLS d'OpenLDAP utilise le mécanisme OLC (On-Line Configuration, cn=config) et les certificats Let's Encrypt générés par Certbot.

```
 # Installer Certbot pour Let's Encrypt
apt install -y certbot

# Générer un certificat pour le serveur LDAP (nécessite un DNS public valide)
certbot certonly --standalone -d ldap.domaine.com --email admin@domaine.com --agree-tos

# Les certificats sont dans : /etc/letsencrypt/live/ldap.domaine.com/
# - fullchain.pem : certificat + chaîne
# - privkey.pem   : clé privée

# Copier les certificats dans /etc/ldap/certs/ (lisibles par l'utilisateur openldap)
mkdir -p /etc/ldap/certs
cp /etc/letsencrypt/live/ldap.domaine.com/fullchain.pem /etc/ldap/certs/
cp /etc/letsencrypt/live/ldap.domaine.com/privkey.pem /etc/ldap/certs/
chown -R openldap:openldap /etc/ldap/certs/
chmod 640 /etc/ldap/certs/privkey.pem

# Configurer TLS dans OpenLDAP via LDIF OLC
cat > /tmp/tls_config.ldif << 'EOF'
dn: cn=config
changetype: modify
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ldap/certs/fullchain.pem
-
add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ldap/certs/fullchain.pem
-
add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ldap/certs/privkey.pem
-
add: olcTLSMinProtocol
olcTLSMinProtocol: 3.3
EOF

ldapmodify -Y EXTERNAL -H ldapi:/// -f /tmp/tls_config.ldif

# Activer LDAPS (port 636) dans /etc/default/slapd
sed -i 's/SLAPD_SERVICES="ldap:\/\/\/ ldapi:\/\/\/"/SLAPD_SERVICES="ldap:\/\/\/ ldaps:\/\/\/ ldap

systemctl restart slapd
```

```
# Tester la connexion LDAPS
ldapsearch -x -H ldaps://ldap.domaine.com -D "cn=admin,dc=domaine,dc=local"     -W -b "dc=domaine
```

> En déploiement d'annuaire OpenLDAP pour une ESN (100 utilisateurs, 15 serveurs Linux), le principal défi a été la migration depuis /etc/passwd locaux vers LDAP centralisé. L'approche retenue : un script Python pour extraire les entrées /etc/passwd et générer les LDIF correspondants, un déploiement SSSD sur tous les serveurs via Ansible, et une période de coexistence de 2 semaines avec les comptes locaux désactivés progressivement après validation LDAP. L'uidNumber doit impérativement être unique dans tout le parc — utiliser un range dédié (ex: 10000-19999) différent des comptes locaux (1000-9999).
>
> *— Retour de mission déploiement LDAP, novembre 2025*

Installation et configuration de phpLDAPadmin

phpLDAPadmin est l'interface web de référence pour administrer OpenLDAP. Développée en PHP, elle permet de naviguer dans l'arbre LDAP, créer et modifier des entrées, gérer les schémas et importer/exporter des fichiers LDIF, sans nécessiter de connaître la syntaxe LDAP en détail.

```
 # Installation de phpLDAPadmin (disponible dans les dépôts Debian)
apt install -y phpldapadmin apache2

# Configuration de phpLDAPadmin
nano /etc/phpldapadmin/config.php

# Modifier les lignes suivantes :
# $config->custom->appearance['hide_template_warning'] = true;
# $servers->setValue('server','host','127.0.0.1');
# $servers->setValue('server','base',array('dc=domaine,dc=local'));
# $servers->setValue('login','bind_id','cn=admin,dc=domaine,dc=local');
# $servers->setValue('server','tls',true);  # Activer TLS

# Sécuriser l'accès à phpLDAPadmin : restreindre aux IP de l'admin
cat > /etc/apache2/conf-available/phpldapadmin.conf << 'EOF'
Alias /phpldapadmin /usr/share/phpldapadmin/htdocs

    Options +FollowSymLinks
    AllowOverride None
    Require ip 10.0.0.0/24 127.0.0.1

EOF

a2enconf phpldapadmin
systemctl reload apache2

# Accès via : http://serveur-ldap/phpldapadmin
# Login : cn=admin,dc=domaine,dc=local + mot de passe admin LDAP
```

Intégration PAM Linux pour l'authentification centralisée

L'intégration LDAP avec PAM Linux permet aux utilisateurs de se connecter à n'importe quel serveur avec leurs identifiants LDAP. La solution moderne recommandée est SSSD (System Security Services Daemon), qui gère le cache local des identités LDAP (important pour les connexions hors ligne) et supporte OpenLDAP, FreeIPA et Active Directory.

```
 # Installation de SSSD et ses dépendances PAM/NSS
apt install -y sssd sssd-ldap libnss-sss libpam-sss

# Configuration SSSD pour OpenLDAP
cat > /etc/sssd/sssd.conf << 'EOF'
[sssd]
services = nss, pam
config_file_version = 2
domains = DOMAINE.LOCAL

[domain/DOMAINE.LOCAL]
id_provider = ldap
auth_provider = ldap
ldap_uri = ldaps://ldap.domaine.com
ldap_search_base = dc=domaine,dc=local
ldap_default_bind_dn = cn=readonly,ou=services,dc=domaine,dc=local
ldap_default_authtok = MotDePasseReadonly2026!
ldap_tls_reqcert = demand
ldap_tls_cacert = /etc/ssl/certs/ca-certificates.crt
ldap_user_search_base = ou=users,dc=domaine,dc=local
ldap_group_search_base = ou=groups,dc=domaine,dc=local
cache_credentials = true
enumerate = false
EOF

chmod 600 /etc/sssd/sssd.conf
chown root:root /etc/sssd/sssd.conf

# Activer et démarrer SSSD
systemctl enable sssd && systemctl start sssd

# Activer la création automatique du répertoire home à la connexion
pam-auth-update --enable mkhomedir

# Test : résoudre un utilisateur LDAP
id alice
# Résultat attendu : uid=10001(alice) gid=10000(sysadmins) groups=10000(sysadmins)

# Test d'authentification LDAP
su - alice
# Entrer le mot de passe LDAP d'alice
```

Sauvegarde et restauration d'OpenLDAP

La sauvegarde de l'annuaire OpenLDAP s'effectue via la commande `slapcat`, qui exporte l'ensemble des entrées en format LDIF sans arrêter le service. Cette approche est à inclure dans les scripts de sauvegarde quotidiens via cron.

```
 # Sauvegarde complète de l'annuaire (sans arrêter slapd)
BACKUP_DIR="/var/backups/ldap"
DATE=$(date +%Y%m%d_%H%M%S)
mkdir -p "$BACKUP_DIR"

slapcat -l "$BACKUP_DIR/ldap_backup_$DATE.ldif"
gzip "$BACKUP_DIR/ldap_backup_$DATE.ldif"

# Nettoyer les sauvegardes de plus de 30 jours
find "$BACKUP_DIR" -name "*.ldif.gz" -mtime +30 -delete

echo "Sauvegarde LDAP terminée : $BACKUP_DIR/ldap_backup_$DATE.ldif.gz"

# Restauration (sur un nouveau serveur avec slapd arrêté)
# 1. Arrêter slapd
systemctl stop slapd

# 2. Supprimer la base de données existante
rm -rf /var/lib/ldap/*

# 3. Restaurer depuis le LDIF
slapadd -l /tmp/ldap_backup_20260115.ldif
chown -R openldap:openldap /var/lib/ldap/

# 4. Redémarrer slapd
systemctl start slapd && systemctl status slapd
```

La documentation officielle d'OpenLDAP est disponible sur openldap.org/doc/admin26/ — section 5 pour la configuration OLC (cn=config) et section 12 pour la sécurité TLS. Pour les aspects d'authentification centralisée et de gestion des identités dans un contexte de sécurité, l'article Forensics Linux : Artifacts et Investigation montre comment analyser les traces d'authentification LDAP lors d'un incident. La gestion des permissions LDAP s'inscrit dans une stratégie plus large d'administration sécurisée des annuaires d'entreprise.

Questions fréquentes

Quelle est la différence entre OpenLDAP et FreeIPA ?

OpenLDAP est uniquement un serveur d'annuaire LDAP — il gère les entrées (utilisateurs, groupes) mais ne fournit pas nativement Kerberos ni la gestion de DNS. FreeIPA (utilisé dans Red Hat Identity Management) combine OpenLDAP, MIT Kerberos, Dogtag PKI et un DNS intégré pour fournir une alternative complète à Active Directory sur Linux. FreeIPA est recommandé pour les environnements Linux purs qui ont besoin de SSO Kerberos ; OpenLDAP seul suffit pour l'authentification simple par login/mot de passe sans ticket Kerberos.

Comment changer le mot de passe d'un utilisateur LDAP depuis la ligne de commande ?

Deux méthodes sont disponibles. La commande `ldappasswd` permet à un administrateur de changer le mot de passe d'un utilisateur : `ldappasswd -x -H ldap://localhost -D "cn=admin,dc=domaine,dc=local" -W -S "uid=alice,ou=users,dc=domaine,dc=local"`. L'option `-S` demande le nouveau mot de passe interactivement. Un utilisateur peut également changer son propre mot de passe en s'authentifiant avec son DN complet et en utilisant `ldappasswd -x -H ldap://localhost -D "uid=alice,ou=users,dc=domaine,dc=local" -W -S`.

OpenLDAP peut-il remplacer Active Directory pour un parc Windows ?

Pas directement sans Samba. Samba 4 (disponible dans les dépôts Debian) intègre OpenLDAP comme backend et implémente les protocoles Active Directory (Kerberos, SMB, Group Policy basique), permettant de joindre des postes Windows à un domaine Samba AD. Cependant, la compatibilité avec les GPO avancées, Azure AD Connect et les fonctionnalités modernes de Windows 11 reste limitée. Pour un parc mixte Windows/Linux important, Active Directory ou Entra ID (Azure AD) restent les solutions recommandées côté Windows.

Comment indexer des attributs LDAP pour améliorer les performances de recherche ?

OpenLDAP supporte plusieurs types d'index (eq, sub, pres, approx) sur les attributs fréquemment utilisés dans les recherches. Les index recommandés pour un annuaire standard : `objectClass eq`, `uid eq`, `uidNumber eq`, `gidNumber eq`, `mail eq`. La configuration OLC via LDIF : `olcDbIndex: uid eq,pres`. Après ajout d'un index, lancer `slapindex` (avec slapd arrêté) pour construire l'index sur les données existantes.

SSSD ou nslcd pour l'intégration PAM LDAP sur Debian ?

SSSD est la solution moderne recommandée depuis Debian 10. Contrairement à `nslcd` (nss-ldapd), SSSD gère le cache local des identités (connexion possible même si le serveur LDAP est temporairement inaccessible), supporte plusieurs fournisseurs d'identité en parallèle (LDAP + AD), offre une meilleure gestion des mots de passe expirés et s'intègre avec sudo via sssd-sudo. `libnss-ldap` et `nslcd` restent utilisables mais sont moins maintenus activement — SSSD est le choix par défaut pour tous les nouveaux déploiements Debian 12.

Gestion des politiques de mot de passe avec le module ppolicy

OpenLDAP supporte les politiques de mot de passe via le module `ppolicy` (Password Policy). Ce module permet d'implémenter des règles de complexité, d'expiration et de verrouillage de comptes directement dans l'annuaire LDAP, sans dépendre des politiques de l'OS hôte. La politique ppolicy s'applique à toutes les authentifications LDAP, qu'elles viennent de PAM Linux, d'applications web ou d'autres clients LDAP.

```
 # Activer le module ppolicy dans OpenLDAP
cat > /tmp/enable_ppolicy.ldif << 'EOF'
dn: cn=module{0},cn=config
changetype: modify
add: olcModuleLoad
olcModuleLoad: ppolicy
EOF

ldapmodify -Y EXTERNAL -H ldapi:/// -f /tmp/enable_ppolicy.ldif

# Créer une politique de mot de passe par défaut
cat > /tmp/ppolicy_default.ldif << 'EOF'
dn: ou=policies,dc=domaine,dc=local
objectClass: organizationalUnit
ou: policies

dn: cn=default,ou=policies,dc=domaine,dc=local
objectClass: pwdPolicy
objectClass: person
objectClass: top
cn: default
sn: default
pwdAttribute: userPassword
pwdMinLength: 12
pwdMaxAge: 7776000
pwdInHistory: 5
pwdMaxFailure: 5
pwdLockout: TRUE
pwdLockoutDuration: 900
EOF

ldapadd -x -H ldap://localhost -D 'cn=admin,dc=domaine,dc=local' \
    -W -f /tmp/ppolicy_default.ldif
```

Réplication OpenLDAP syncrepl pour la haute disponibilité

La réplication OpenLDAP (syncrepl) assure la haute disponibilité de l'annuaire avec un provider (maître) et un ou plusieurs consumers (répliques). En cas de défaillance du provider, les consumers continuent à répondre aux requêtes de lecture, garantissant la continuité d'authentification pour les serveurs Linux qui dépendent de SSSD. La réplication syncrepl fonctionne en mode refreshAndPersist (pull permanent) ou refreshOnly (pull périodique).

```
 # Activer la réplication sur le serveur PROVIDER
cat > /tmp/sync_provider.ldif << 'EOF'
dn: cn=module{0},cn=config
changetype: modify
add: olcModuleLoad
olcModuleLoad: syncprov

dn: olcOverlay={0}syncprov,olcDatabase={1}mdb,cn=config
objectClass: olcOverlayConfig
objectClass: olcSyncProvConfig
olcOverlay: syncprov
olcSpCheckpoint: 100 10
olcSpSessionLog: 100
EOF

ldapmodify -Y EXTERNAL -H ldapi:/// -f /tmp/sync_provider.ldif

# Configuration du CONSUMER (réplique)
cat > /tmp/sync_consumer.ldif << 'EOF'
dn: olcDatabase={1}mdb,cn=config
changetype: modify
add: olcSyncRepl
olcSyncRepl: rid=001
  provider=ldap://ldap-provider.domaine.local
  type=refreshAndPersist
  retry="5 5 300 +"
  searchbase="dc=domaine,dc=local"
  attrs="*,+"
  bindmethod=simple
  binddn="cn=admin,dc=domaine,dc=local"
  credentials=MotDePasseAdmin
EOF

ldapmodify -Y EXTERNAL -H ldapi:/// -f /tmp/sync_consumer.ldif
```

Monitoring OpenLDAP avec cn=Monitor

OpenLDAP intègre un backend de monitoring accessible via le DN `cn=Monitor`. Il expose des statistiques en temps réel : connexions actives, opérations LDAP par type (search, bind, modify), bytes transmis et waiters (requêtes en attente de traitement). Ces métriques sont exportables vers Prometheus via `prometheus-openldap-exporter` pour des dashboards Grafana.

```
 # Consulter les statistiques de monitoring
ldapsearch -x -H ldap://localhost -D 'cn=admin,dc=domaine,dc=local' \
    -W -b 'cn=Monitor' '(objectClass=monitorCounterObject)' \
    monitorCounter cn | grep -E '^(cn|monitorCounter):'

# Connexions actives
ldapsearch -Y EXTERNAL -H ldapi:/// \
    -b 'cn=Current,cn=Connections,cn=Monitor' \
    monitorCounter 2>/dev/null | grep monitorCounter

# Nombre d'opérations BIND depuis le démarrage
ldapsearch -Y EXTERNAL -H ldapi:/// \
    -b 'cn=Bind,cn=Operations,cn=Monitor' \
    monitorOpInitiated monitorOpCompleted 2>/dev/null
```

Pour approfondir la gestion des annuaires, la documentation officielle OpenLDAP sur la réplication détaille les modes push, pull et miroir. La supervision d'OpenLDAP s'intègre dans une démarche de détection des persistances sur les systèmes Linux et de surveillance proactive de l'infrastructure Active Directory et LDAP.

Schéma LDAP : ajouter des attributs personnalisés

Le schéma LDAP définit les objectClasses et attributs disponibles dans l'annuaire. OpenLDAP inclut les schémas standard (core, inetOrgPerson, posixAccount, shadowAccount), mais il est possible d'ajouter des schémas personnalisés pour des besoins métier spécifiques : numéro de badge, département interne, droits applicatifs. La définition de schéma personnalisé utilise un OID (Object Identifier) unique pour chaque attribut et objectClass.

```
 # Créer un schéma LDAP personnalisé (example : attributs RH)
cat > /tmp/custom_schema.ldif << 'EOF'
dn: cn=custom-schema,cn=schema,cn=config
objectClass: olcSchemaConfig
cn: custom-schema
# Définir un attribut personnalisé (OID fictif à remplacer par un OID IANA réel)
olcAttributeTypes: ( 1.3.6.1.4.1.99999.1.1
  NAME 'employeeId'
  DESC 'Identifiant unique employé'
  EQUALITY caseExactMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{64}
  SINGLE-VALUE )
# Définir une objectClass qui contient l'attribut
olcObjectClasses: ( 1.3.6.1.4.1.99999.2.1
  NAME 'employeeInfo'
  DESC 'Informations RH de l employé'
  AUXILIARY
  MAY ( employeeId ) )
EOF

ldapadd -Y EXTERNAL -H ldapi:/// -f /tmp/custom_schema.ldif

# Ajouter l'attribut à un utilisateur existant
cat > /tmp/add_employeeid.ldif << 'EOF'
dn: uid=alice,ou=users,dc=domaine,dc=local
changetype: modify
add: objectClass
objectClass: employeeInfo
-
add: employeeId
employeeId: EMP-2026-00142
EOF

ldapmodify -x -H ldap://localhost -D 'cn=admin,dc=domaine,dc=local' \
    -W -f /tmp/add_employeeid.ldif

# Vérifier
ldapsearch -x -H ldap://localhost -D 'cn=admin,dc=domaine,dc=local' \
    -W -b 'ou=users,dc=domaine,dc=local' '(employeeId=EMP-2026-00142)' \
    uid employeeId
```

Accès LDAP read-only pour les applications

Les applications qui authentifient les utilisateurs via LDAP (Nextcloud, GitLab, Zabbix, phpIPAM, Grafana) n'ont besoin que d'un accès lecture sur l'annuaire. Créer un compte de service read-only avec des droits strictement limités à la recherche d'utilisateurs est une bonne pratique de sécurité — si les credentials de l'application sont compromis, l'attaquant ne peut que lire l'annuaire, pas le modifier.

```
 # Créer un compte de service read-only pour les applications
cat > /tmp/service_readonly.ldif << 'EOF'
dn: uid=svc-readonly,ou=services,dc=domaine,dc=local
objectClass: inetOrgPerson
objectClass: posixAccount
uid: svc-readonly
sn: Service
givenName: ReadOnly
cn: Service ReadOnly
uidNumber: 20001
gidNumber: 20000
homeDirectory: /nonexistent
loginShell: /bin/false
userPassword: {SSHA}HASH_MOT_DE_PASSE_FORT
description: Compte de service lecture seule pour applications
EOF

ldapadd -x -H ldap://localhost -D 'cn=admin,dc=domaine,dc=local' \
    -W -f /tmp/service_readonly.ldif

# Configurer les ACLs pour limiter les droits du compte read-only
# Le compte peut lire uid, cn, mail, memberOf — pas userPassword
cat > /tmp/acl_readonly.ldif << 'EOF'
dn: olcDatabase={1}mdb,cn=config
changetype: modify
add: olcAccess
olcAccess: {0}to attrs=userPassword
  by dn.exact="cn=admin,dc=domaine,dc=local" write
  by self write
  by anonymous auth
  by * none
olcAccess: {1}to dn.subtree="ou=users,dc=domaine,dc=local"
  by dn.exact="uid=svc-readonly,ou=services,dc=domaine,dc=local" read
  by dn.exact="cn=admin,dc=domaine,dc=local" write
  by self read
  by * none
EOF

ldapmodify -Y EXTERNAL -H ldapi:/// -f /tmp/acl_readonly.ldif
```

L'annuaire LDAP OpenLDAP ainsi configuré constitue le socle d'une gestion des identités robuste pour les environnements Linux. Pour une gestion centralisée des identités à plus grande échelle incluant Kerberos, consulter l'article sur les artifacts d'authentification Linux utiles lors

d'investigations forensiques. La sécurisation de l'accès LDAP s'inscrit dans la politique de gestion des identités et accès (IAM) préconisée par le cadre de sécurisation des annuaires d'entreprise qui couvre aussi bien OpenLDAP que Active Directory.

Cas d'usage : centraliser l'authentification SSH avec OpenLDAP et SSSD

L'un des cas d'usage les plus courants d'OpenLDAP dans les infrastructures Linux est la centralisation de l'authentification SSH. Avec SSSD configuré sur tous les serveurs, un administrateur peut créer un compte une seule fois dans l'annuaire LDAP et se connecter immédiatement à tous les serveurs du parc, sans copier manuellement les clés SSH ni synchroniser les fichiers `/etc/passwd`. La gestion des clés SSH publiques dans l'annuaire LDAP (via l'attribut `sshPublicKey` de l'objectClass `ldapPublicKey`) complète cette intégration : l'administrateur SSH Authorized Keys est remplacé par une lecture LDAP.

```
 # Configurer SSH pour lire les clés publiques depuis LDAP
# Installer le module SSH LDAP
apt install -y libpam-ldap ldap-utils

# Ajouter le schéma openssh-lpk
ldapadd -Y EXTERNAL -H ldapi:/// \
    -f /usr/share/doc/ldap-utils/examples/openssh-lpk.ldif

# Ajouter la clé SSH publique d'un utilisateur dans LDAP
cat > /tmp/add_sshkey.ldif << 'EOF'
dn: uid=alice,ou=users,dc=domaine,dc=local
changetype: modify
add: objectClass
objectClass: ldapPublicKey
-
add: sshPublicKey
sshPublicKey: ssh-ed25519 AAAA...clé_publique_alice... alice@poste
EOF

ldapmodify -x -H ldap://localhost -D 'cn=admin,dc=domaine,dc=local' \
    -W -f /tmp/add_sshkey.ldif

# Configurer sshd pour lire les clés depuis LDAP
# Dans /etc/ssh/sshd_config :
# AuthorizedKeysCommand /usr/bin/sss_ssh_authorizedkeys
# AuthorizedKeysCommandUser nobody
systemctl reload sshd
```

Cette configuration d'authentification SSH centralisée via LDAP est documentée dans les guides de durcissement CIS (Center for Internet Security) pour Linux. Pour les aspects forensiques des connexions SSH et des authentifications LDAP lors d'investigations de sécurité, consulter l'article Forensics Linux : artifacts et investigation.