

# Document Complet : LDAP vs SCRAM + Commandes Kafka + API REST

Kafka 4.0 avec KRaft renforce la sécurité et la gestion des identités. Ce document compare LDAP et SCRAM, inclut des schémas, une analyse avancée et toutes les commandes Kafka (users, ACL, API REST).

## Architecture : SSL/SASL + LDAP

Client → SSL → Broker

SASL/PLAIN ou GSSAPI → LDAP

Identités centralisées AD/LDAP

Kafka délègue l'authentification

## Architecture : SSL/SASL + SCRAM

Client → SSL → Broker

SASL/SCRAM interne Kafka

Identités SCRAM stockées dans le cluster

Kafka autonome pour authentification

## Analyse approfondie

### 1. SSL/SASL + LDAP

Cette approche est généralement utilisée dans les grandes organisations disposant d'un Active Directory ou d'un annuaire LDAP centralisé. Elle permet à Kafka de ne pas gérer lui-même les identités : celles-ci restent dans un service déjà en place, souvent connecté à d'autres systèmes internes.

#### Avantages :

- Intégration native aux politiques internes de sécurité et conformité.
- Désactivation automatique d'un utilisateur dans tout le SI (déprovisionnement global).
- Pas besoin de créer des comptes spécifiques Kafka.
- Support de mécanismes d'authentification avancés (Kerberos/GSSAPI).

#### Inconvénients :

- Dépendance forte à un service externe : si LDAP tombe, les connexions SASL échouent.
- Complexité accrue de configuration (certificats + SASL + LDAP).
- Risque de latence en cas de volumétrie élevée d'authentification.
- Les administrateurs Kafka n'ont pas de visibilité directe sur les identifiants.

### 2. SSL/SASL + SCRAM

SCRAM est un mécanisme plus moderne, léger et intégré directement dans Kafka. Les comptes utilisateurs sont stockés dans le cluster, ce qui simplifie l'architecture et augmente les performances.

#### Avantages :

- Très performant, idéal pour les clusters à très forte charge.
- Aucune dépendance à un système externe.
- Configuration plus simple à maintenir.
- Bon compromis entre sécurité et administration.

#### Inconvénients :

- Les mots de passe doivent être gérés manuellement (création, rotation).
- Pas de synchronisation automatique avec AD/LDAP.
- Moins adapté aux organisations avec forte gouvernance IAM.

## Conclusion

Le choix entre LDAP et SCRAM dépend principalement du contexte organisationnel :

- **LDAP** : idéal pour les entreprises souhaitant centraliser la gestion des identités.
- **SCRAM** : parfait pour des environnements techniques recherchant performance, autonomie et simplicité.

Dans Kafka KRaft, SCRAM devient très populaire grâce à sa rapidité et son intégration native, tandis que LDAP reste essentiel dans les organisations exigeant un contrôle RH/IAM strict.

## **Création d'un utilisateur SCRAM**

```
kafka-configs --bootstrap-server broker1:9092 \
--alter \
--add-config 'SCRAM-SHA-512=[password=monpassword]' \
--entity-type users \
--entity-name producer-app
```

## **ACL Producer**

```
kafka-acls --bootstrap-server broker1:9092 \
--add \
--allow-principal User:producer-app \
--operation Write \
--topic orders
```

## **ACL Consumer**

```
kafka-acls --bootstrap-server broker1:9092 \
--add \
--allow-principal User:consumer-app \
--operation Read \
--topic orders

kafka-acls --bootstrap-server broker1:9092 \
--add \
--allow-principal User:consumer-app \
--operation Read \
--group consumer-orders-group
```

## **API REST Confluent – Crédation d'un user via REST**

```
curl -X PUT \
-u admin:admin-secret \
-H "Content-Type: application/json" \
--data '{ "password": "monpassword" }' \
https://kafka-broker:8091/v1/users/producer-app
```