Seafile

Installation

Parfait, on plonge dans **Seafile** en détail ! Voici tout ce qu'il faut savoir pour **maîtriser Seafile** : installation avancée, optimisation, gestion des utilisateurs, sauvegardes, et astuces pour une utilisation fluide en équipe. Je vais te donner des éléments concrets, des exemples de commandes, et des retours d'expérience pour éviter les pièges.

---

1. Architecture et Concepts Clés de Seafile

a. Comment Seafile organise les données ?

Seafile utilise un système de **blocs de fichiers** (comme Git) :

- Chaque fichier est découpé en **blocs de 1 Mo** (par défaut).
- Les blocs sont **hashés** (SHA-1) et stockés dans `/shared/seafile-data`.
- Les métadonnées (noms, permissions) sont stockées dans la **base de données MariaDB** .

Avantages :

- **Économie d'espace** : Les blocs identiques (ex : fichiers dupliqués) ne sont stockés qu'une fois.
- **Synchro rapide** : Seuls les blocs modifiés sont transférés.
- **Intégrité des données** : Les hashs permettent de détecter les corruptions.

Exemple de structure de dossiers :

```
/srv/seafile/data/
├── seafile-data/       # Blocs de fichiers
├── seafile-conf/       # Configurations
└── logs/               # Logs du serveur
```

---

b. Ports utilisés par Seafile

Port	Usage	À exposer ?
8000	Interface web	Oui (via Nginx)

Port	Usage	À exposer ?
8082	API pour le client desktop	Oui (si synchro hors réseau local)
10001	Service de notification	Non (interne)
12001	Service de recherche	Non (interne)

Conseil : Dans ton Nginx, expose **uniquement le port 8000** (le reste est géré en interne par Docker).

---

2. Optimisation des Performances

a. Configuration de Seafile pour Docker

Pour optimiser les performances, ajoute ces variables d'environnement dans ton `docker-compose.yml` :

```
environment:
  - SEAFILE_SERVER_HOSTNAME=seafile.tacampagne.fr
  - SEAFILE_ADMIN_EMAIL=admin@tacampagne.fr
  - SEAFILE_ADMIN_PASSWORD=tonmotdepasse
  - SEAFILE_MYSQL_HOST=seafile-db
  - SEAFILE_MYSQL_ROOT_PASSWORD=motdepassebdd
  - SEAFILE_TIMEZONE=Europe/Paris  # Pour les horodatages
  - SEAFILE_FILE_SERVER_ROOT=https://seafile.tacampagne.fr  # URL publique
  - SEAFILE_ENABLE_FILE_LOCKING=true  # Évite les conflits d'édition
```

b. Tuning de la base de données (MariaDB)

Ajoute ces lignes dans la section `seafile-db` de ton `docker-compose.yml` pour améliorer les performances :

```
command: [
  '--character-set-server=utf8mb4',
  '--collation-server=utf8mb4_unicode_ci',
  '--innodb-buffer-pool-size=512M',  # Augmente si tu as beaucoup de fichiers
  '--max-connections=200'
]
```

c. Configuration Nginx pour les gros fichiers

Si tu partages des fichiers > 100 Mo, ajoute ces directives dans ton bloc `server` Nginx :

```
proxy_connect_timeout       300;
proxy_send_timeout          300;
```

```
proxy_read_timeout          300;
send_timeout                300;
client_max_body_size        10G;  # Autorise les uploads jusqu'à 10 Go
```

---

3. Gestion des Utilisateurs et Permissions

a. Créer des utilisateurs en masse

Seafile permet d'importer des utilisateurs via un **fichier CSV**. Exemple de format :

```
Email,Password,Name,Quota (in MB)
user1@tacampagne.fr,motdepasse1,Jean Dupont,1024
user2@tacampagne.fr,motdepasse2,Marie Martin,1024
```

Commande pour importer (à exécuter dans le conteneur Seafile) :

```
docker exec -it seafile bash
cd /opt/seafile/seafile-server-latest
./seafile-admin.sh import-users user_list.csv
```

b. Gérer les groupes

- **Crée un groupe** pour ton équipe :
  ```
  docker exec -it seafile bash
  cd /opt/seafile/seafile-server-latest
  ./seafile-admin.sh create-group "Equipe Campagne"
  ```
- **Ajoute des utilisateurs au groupe** :
  ```
  ./seafile-admin.sh add-group-member "Equipe Campagne" user1@tacampagne.fr
  ```

c. Partager un dossier avec un groupe

1. Connecte-toi à l'interface web (`https://seafile.tacampagne.fr`).
2. Crée un dossier (ex: `Documents Campagne`).
3. Clique sur **Partager** → Sélectionne le groupe **"Equipe Campagne"** → Choisis les permissions (**Lecture/Ecriture**).

---

4. Sauvegardes et Restauration

a. Sauvegarde complète (données + DB)

Script de sauvegarde (`/srv/seafile/backup.sh`) :

```
#!/bin/bash
BACKUP_DIR="/srv/backups/seafile-$(date +%Y-%m-%d)"
mkdir -p "$BACKUP_DIR"

# Sauvegarde des données
docker-compose -f /srv/seafile/docker-compose.yml exec seafile tar -czvf /tmp/seafile-data.tar.gz /shared
docker cp seafile:/tmp/seafile-data.tar.gz "$BACKUP_DIR"

# Sauvegarde de la base de données
docker-compose -f /srv/seafile/docker-compose.yml exec seafile-db mysqldump -uroot -pmotdepassebdd --all-databases > "$BACKUP_DIR/seafile-db.sql"

# Nettoyage
docker-compose -f /srv/seafile/docker-compose.yml exec seafile rm /tmp/seafile-data.tar.gz
```

Automatise avec cron :

```
0 2 * * * /bin/bash /srv/seafile/backup.sh  # Sauvegarde tous les jours à 2h
```

b. Restauration

1. **Restaure les données** :
   `docker cp "$BACKUP_DIR/seafile-data.tar.gz" seafile:/tmp`
   `docker-compose -f /srv/seafile/docker-compose.yml exec seafile tar -xzvf /tmp/seafile-data.tar.gz -C /`
2. **Restaure la DB** :
   `cat "$BACKUP_DIR/seafile-db.sql" | docker-compose -f /srv/seafile/docker-compose.yml exec -T seafile-db mysql -uroot -pmotdepassebdd`
3. **Redémarre les conteneurs** :
   `docker-compose -f /srv/seafile/docker-compose.yml restart`

---

5. Sécurité Avancée

a. Activer le chiffrement côté serveur

Seafile propose un **chiffrement des fichiers au repos** (server-side encryption) :

1. Active-le dans `docker-compose.yml` :
   ```
   environment:
     - SEAFILE_ENABLE_ENCRYPTION=true
   ```
2. **Chiffre un dossier existant** :

- Va dans l'interface web → Sélectionne un dossier → **Propriétés** → **Activer le chiffrement**.

Attention : Si tu perds la clé de chiffrement, les données sont **perdues à jamais**.

b. Restreindre l'accès par IP

Dans ton Nginx, ajoute une **liste blanche d'IPs** pour limiter l'accès :

```
location / {
    allow 192.168.1.0/24;  # Remplace par ton réseau local ou les IPs de ton équipe
    allow 82.123.45.67;    # IP fixe d'un membre
    deny all;
    proxy_pass http://<IP_SERVEUR_A>:8000;
    ...
}
```

c. Activer la double authentification (2FA)

1. Dans l'interface admin (`https://seafile.tacampagne.fr/sysadmin/`), va dans **Paramètres → 2FA**.
2. Active **Google Authenticator** ou **Email OTP**.
3. Demande à tes utilisateurs de configurer le 2FA via leur profil.

---

6. Intégration avec d'Outils Externes

a. Synchronisation avec le client desktop

- **Télécharge le client** : Seafile Desktop (Windows/macOS/Linux).
- **Configuration** :
  - URL du serveur : `https://seafile.tacampagne.fr`
  - Email/mot de passe de l'utilisateur.
- **Options utiles** :
  - **Synchro sélective** : Pour ne pas tout télécharger.
  - **Verrouillage de fichiers** : Évite les conflits si 2 personnes éditent le même fichier.

b. Montage en WebDAV (pour les power users)

Seafile supporte le **WebDAV** pour monter le stockage comme un disque réseau :

- **URL WebDAV** : `https://seafile.tacampagne.fr/seafdav`
- **Exemple de montage sous Linux** :
  ```
  sudo apt install davfs2
  echo "https://seafile.tacampagne.fr/seafdav /mnt/seafile davfs user,noauto 0 0" >> /etc/fstab
  mount /mnt/seafile
  ```
  (L'utilisateur devra entrer son email/mot de passe.)

---

7. Monitoring et Logs

a. Vérifier les logs

- **Logs Seafile** :
  `docker-compose -f /srv/seafile/docker-compose.yml logs -f seafile`
- **Logs Nginx** :
  `tail -f /var/log/nginx/seafile.tacampagne.fr-access.log`

b. Surveiller l'espace disque

`docker exec seafile du -sh /shared/seafile-data`

c. Alertes par email

Configure Seafile pour envoyer des alertes (ex : espace disque faible) :

1. Dans l'interface admin, va dans **Paramètres → Email**.
2. Configure un serveur SMTP (ex : ton serveur mail ou Gmail).
3. Active les notifications pour les **quotas** et les **erreurs système**.

---

8. Astuces pour les Utilisateurs

a. Partager un fichier avec un externe

1. **Crée un lien public** :
   - Clique droit sur un fichier/dossier → **Partager** → **Créer un lien**.
   - Définis une **date d'expiration** et un **mot de passe** si besoin.
2. **Envoie le lien** par email ou messagerie.

b. Récupérer un fichier supprimé

- Seafile garde une **corbeille** (30 jours par défaut).
- Pour restaurer : **Corbeille** → Sélectionne le fichier → **Restaurer**.

c. Utiliser l'API

Seafile propose une API REST pour automatiser des tâches (ex : uploads depuis un script Python). Exemple pour lister les dossiers :

```
curl -u admin@tacampagne.fr:tonmotdepasse https://seafile.tacampagne.fr/api2/repos/
```

---

9. Résolution des Problèmes Courants

Problème	Solution
Erreur 502 Bad Gateway (Nginx)	Vérifie que le conteneur Seafile est démarré (`docker ps`).
Synchro bloquée	Redémarre le client desktop ou vérifie les logs (`docker logs seafile`).
Quota dépassé	Augmente le quota via l'interface admin ou supprime des fichiers.
Lien public inaccessible	Vérifie que le port `8000` est bien exposé et que Nginx proxy correctement.

---

10. Pour Aller Plus Loin

a. High Availability (HA)

Si tu veux une **redondance** (ex : 2 serveurs Seafile) :

- Utilise **Seafile Pro** (version payante) avec support HA.
- Ou configure un **réplica de la base de données** + un partage NFS pour `/shared/seafile-data`.

b. Intégration avec OnlyOffice

Si tu veux éditer des docs en ligne (comme Nextcloud) :

1. Déploie **OnlyOffice** en conteneur Docker.
2. Configure l'intégration dans Seafile :
   - Dans l'admin, va dans **Paramètres → OnlyOffice**.
   - Renseigne l'URL de ton instance OnlyOffice.

---

11. Ressources Utiles

- **Documentation officielle** : Seafile Manual
- **Forum communautaire** : Seafile Forum
- **Dépôt Docker** : Seafile sur Docker Hub

---

12. Prochaines Étapes

1. **Veux-tu que je te génère un script complet pour les sauvegardes incrémentielles (avec rotation) ?**
2. **As-tu besoin d'aide pour configurer OnlyOffice ou un autre outil intégré ?**
3. **Veux-tu un exemple de politique de sécurité (mots de passe, 2FA, etc.) pour ton équipe ?**

Dis-moi ce qui t'intéresse en priorité, et on creuse ! □□

API et autres

Pas de problème, on reste **minimaliste et efficace** : juste le partage de fichiers/dossiers, avec une intégration simple pour gérer les utilisateurs depuis ton app existante.

---

1. Intégration des Utilisateurs via l'API REST de Seafile

Seafile propose une **API REST complète** pour gérer les utilisateurs, les groupes et les permissions. C'est la solution la plus propre pour automatiser la création de comptes depuis ton app.

a. Documentation de l'API

- **Lien officiel** : Seafile API v2.1
- **Authentification** : Utilise un **token admin** (à générer depuis l'interface web).

---

b. Étapes pour Utiliser l'API

Étape 1 : Générer un Token Admin

1. Connecte-toi à Seafile en tant qu'admin (`https://seafile.tacampagne.fr`).
2. Va dans **Profil → Générer un token API**.
3. Copie ce token (ex: `abc123...xyz`), il servira pour toutes les requêtes API.

Étape 2 : Créer un Utilisateur via l'API

Exemple en `curl` :

```
curl -v -H "Authorization: Token abc123...xyz" \
     -H "Content-Type: application/json" \
     -d '{"email": "user1@tacampagne.fr", "password": "motdepasse123", "name": "Jean Dupont"}' \
     https://seafile.tacampagne.fr/api2/accounts/
```

- **Réponse réussie** :
  `{"email": "user1@tacampagne.fr", "name": "Jean Dupont"}`

Étape 3 : Créer un Groupe et Ajouter des Utilisateurs

1. **Créer un groupe** (ex: "Membres Campagne") :
   ```
   curl -v -H "Authorization: Token abc123...xyz" \
        -H "Content-Type: application/json" \
        -d '{"name": "Membres Campagne"}' \
        https://seafile.tacampagne.fr/api2/groups/
   ```
2. **Ajouter un utilisateur au groupe** :
   ```
   curl -v -H "Authorization: Token abc123...xyz" \
        -X PUT \
   
   https://seafile.tacampagne.fr/api2/groups/Membres%20Campagne/members/user1@tacampagne.fr/
   ```

Étape 4 : Partager un Dossier avec un Groupe

1. **Récupérer l'ID de la bibliothèque (library)** :
   ```
   curl -H "Authorization: Token abc123...xyz" \
        https://seafile.tacampagne.fr/api2/repos/
   ```
   - Exemple de réponse :
     ```
     [
       {
         "id": "a1b2c3d4-5678-90ef-ghij-klmnopqrstuv",
         "name": "Documents Campagne",
         "permission": "rw"
       }
     ]
     ```
2. **Partager le dossier avec le groupe** :
   ```
   curl -v -H "Authorization: Token abc123...xyz" \
        -H "Content-Type: application/json" \
        -d '{"share_type": "group", "group": "Membres Campagne", "permission": "rw"}' \
        https://seafile.tacampagne.fr/api2/repos/a1b2c3d4-5678-90ef-ghij-
   klmnopqrstuv/shares/
   ```

---

c. Exemple en Python (pour ton app)

Si ton app est en Python, voici un script pour créer des utilisateurs depuis une liste :

```
import requests
import json

# Config
SEAFILE_URL = "https://seafile.tacampagne.fr"
API_TOKEN = "abc123...xyz"  # Remplace par ton token
HEADERS = {"Authorization": f"Token {API_TOKEN}", "Content-Type": "application/json"}

# Liste des utilisateurs (ex: extraite de ton app)
users = [
    {"email": "user1@tacampagne.fr", "password": "motdepasse1", "name": "Jean Dupont"},
    {"email": "user2@tacampagne.fr", "password": "motdepasse2", "name": "Marie Martin"}
]

# Créer les utilisateurs
for user in users:
```

```
    response = requests.post(
        f"{SEAFILE_URL}/api2/accounts/",
        headers=HEADERS,
        data=json.dumps(user)
    )
    if response.status_code == 200:
        print(f"Utilisateur {user['email']} créé avec succès.")
    else:
        print(f"Erreur pour {user['email']}: {response.text}")

# Créer un groupe et ajouter les utilisateurs
group_name = "Membres Campagne"
requests.post(
    f"{SEAFILE_URL}/api2/groups/",
    headers=HEADERS,
    data=json.dumps({"name": group_name})
)

for user in users:
    requests.put(
        f"{SEAFILE_URL}/api2/groups/{group_name}/members/{user['email']}/",
        headers=HEADERS
    )
```

---

2. Alternative : Synchronisation via Base de Données

Si tu préfères éviter l'API (ou si ton app tourne sur le même serveur), tu peux **interagir directement avec la base de données MariaDB** de Seafile. **Attention** : Cette méthode est **moins recommandée** (risque de corruption si mal utilisée), mais elle est possible.

a. Structure de la Base de Données

- **Table** `EmailUser` : Stocke les utilisateurs.
  `SELECT * FROM EmailUser;`

Champ	Description
email	Email de l'utilisateur
password	Mot de passe (hashé)
name	Nom complet
is_staff	1 = admin, 0 = utilisateur normal

- **Table** `GroupUser` : Stocke les groupes.
- **Table** `GroupUser_Group` : Lie les utilisateurs aux groupes.

b. Exemple de Requête SQL

Pour ajouter un utilisateur :

```
INSERT INTO EmailUser (email, password, name, is_staff)
VALUES ('user1@tacampagne.fr', 'motdepasse_hashé', 'Jean Dupont', 0);
```

Problème : Le mot de passe doit être **hashé** (Seafile utilise un algorithme spécifique). Pour le générer, utilise l'API ou le script Python ci-dessus.

---

3. Automatisation avec un Script Bash

Si tu veux un script **simple et robuste** pour synchroniser tes utilisateurs depuis un fichier CSV (ex: exporté depuis ton app) :

```
#!/bin/bash
API_TOKEN="abc123...xyz"
SEAFILE_URL="https://seafile.tacampagne.fr"
CSV_FILE="users.csv"  # Format: email,password,name

while IFS=, read -r email password name; do
  # Créer l'utilisateur
  curl -s -H "Authorization: Token $API_TOKEN" \
       -H "Content-Type: application/json" \
       -d "{\"email\": \"$email\", \"password\": \"$password\", \"name\": \"$name\"}" \
       "$SEAFILE_URL/api2/accounts/" | grep -q "email"
  if [ $? -eq 0 ]; then
    echo "Utilisateur $email créé."
    # Ajouter au groupe "Membres Campagne"
    curl -s -X PUT -H "Authorization: Token $API_TOKEN" \
         "$SEAFILE_URL/api2/groups/Membres%20Campagne/members/$email/"
  else
    echo "Erreur pour $email."
  fi
done < "$CSV_FILE"
```

---

4. Bonnes Pratiques

1. **Gère les mots de passe** :
   - Génère des mots de passe aléatoires pour les utilisateurs et envoie-les par email (ex: avec `sendmail` ou un service comme Mailgun).
   - Ou utilise un **lien d'invitation** (Seafile permet d'envoyer des invitations par email).

2. **Sécurise ton token API** :
   - Ne le stocke **jamais en clair** dans ton code. Utilise des variables d'environnement ou un gestionnaire de secrets (ex: `dotenv`).
   - Restreins les permissions du token (ex: uniquement création d'utilisateurs).
3. **Logs et monitoring** :
   - Active les logs de l'API Seafile pour déboguer : `docker-compose logs -f seafile | grep api2`
4. **Teste en staging** :
   - Déploie une instance Seafile de test pour valider ton script avant de l'exécuter en production.

---

5. Exemple Complet : Intégration avec ton App

Scénario :

- Ton app gère une liste de contacts (ex: `contacts.csv`).
- Tu veux :
  1. Créer un utilisateur Seafile pour chaque contact.
  2. Les ajouter au groupe "Membres Campagne".
  3. Partager un dossier "Documents Campagne" avec ce groupe.

Étapes :

1. **Exporte tes contacts** depuis ton app au format CSV :
   ```
   email,password,name
   user1@tacampagne.fr,motdepasse1,Jean Dupont
   user2@tacampagne.fr,motdepasse2,Marie Martin
   ```
2. **Exécute le script Python** (ou Bash) pour créer les utilisateurs et les groupes.
3. **Partage le dossier** via l'API (comme montré plus haut).

---

6. Résumé des Commandes Utiles

Action	Commande curl
Lister les utilisateurs	`curl -H "Authorization: Token ABC" "$SEAFILE_URL/api2/accounts/"`
Créer un utilisateur	`curl -H "Authorization: Token ABC" -d '{"email": "...", "password": "...", ...}' "$SEAFILE_URL/api2/accounts/"`

Action	Commande curl
Créer un groupe	`curl -H "Authorization: Token ABC" -d '{"name": "Groupe"}' "$SEAFILE_URL/api2/groups/"`
Ajouter un utilisateur à un groupe	`curl -X PUT -H "Authorization: Token ABC" "$SEAFILE_URL/api2/groups/Groupe/members/email/"`
Partager un dossier	`curl -H "Authorization: Token ABC" -d '{"share_type": "group", "group": "Groupe", "permission": "rw"}' "$SEAFILE_URL/api2/repos/ID_DU_DOSSIER/shares/"`

---

7. Prochaines Étapes

1. **Veux-tu que je t’aide à adapter le script Python/Bash pour ton format de données spécifique ?**
2. **As-tu besoin d’un exemple pour générer des mots de passe aléatoires et les envoyer par email ?**
3. **Veux-tu tester l’API en direct avec un exemple de `curl` pour vérifier que tout fonctionne ?**

Dis-moi ce qui te serait le plus utile, et on avance ! □□

nouveautés de la v13

● Bonne question — c'est une des choses les plus déroutantes de Seafile v13. Il y a en réalité trois niveaux d'identité distincts :

Identifiant d'utilisateur (UUID@auth.local)

- L'identifiant interne généré automatiquement par Seafile à la création du compte

- Immuable, utilisé par toutes les API et la DB

- Le suffixe @auth.local = compte local (pas LDAP, pas OAuth externe)

E-mail de contact

- Ton vrai email (nicolas.farrie@gmail.com)

- Utilisé pour les notifications, invitations de partage, emails système

- C'est celui qu'on voit affiché partout dans l'UI

- Modifiable

Identifiant de connexion (login_id)
- Un alias optionnel pour se connecter à la place de l'UUID interne

- Quand il est -- (vide), Seafile fait une recherche par email de contact pour le login web — c'est pour ça que tu peux te connecter avec nicolas.farrie@gmail.com sur le web
- L'API auth-token elle, ne fait pas cette recherche — elle cherche directement l'identifiant interne, d'où l'échec systématique

En pratique : si tu voulais que l'API auth-token fonctionne avec nicolas.farrie@gmail.com, il faudrait setter le login_id à cette valeur pour ce compte. Mais vu qu'on passe par les tokens DB

directement, c'est sans importance pour l'instant.