

# 9 - Migration complète d'Immich

Inventorier la source, transférer les données et migrer proprement vers une nouvelle installation.

- [9.1 - Inventaire et plan de migration](#)
- [9.2 - Préparer et exporter le serveur source](#)
- [9.3 - Transférer et préparer la destination](#)
- [9.4 - Cas pratique Proxmox vers TrueNAS](#)

# 9.1 - Inventaire et plan de migration

Une migration réussie commence par un inventaire. Ne pas arrêter ou supprimer la source tant que la destination et la restauration ne sont pas validées.

## Informations à relever

| Élément                | Exemple                        |
|------------------------|--------------------------------|
| Version Immich source  | v2.7.5                         |
| Type de déploiement    | VM Proxmox avec Docker Compose |
| Version PostgreSQL     | valeur du conteneur            |
| Extension vectorielle  | pgvecto.rs ou VectorChord      |
| Emplacement des médias | UPLOAD_LOCATION                |
| Emplacement PostgreSQL | DB_DATA_LOCATION               |
| Bibliothèques externes | chemins hôte et conteneur      |
| Domaine                | immich.dreamsaphir.net         |
| OIDC                   | auth.dreamsaphir.net           |
| Destination            | TrueNAS et dataset Immich      |

## Commandes source

```
cd /chemin/compose/immich
docker compose ps
docker compose images
docker exec immich_server immich-admin version
docker exec immich_server immich-admin schema-check
grep -E '^(UPLOAD_LOCATION|DB_DATA_LOCATION|IMMICH_VERSION|DB_USERNAME|DB_DATABASE_NAME)= '
.env
```

```
docker inspect immich_server --format '{{range .Mounts}}{{println .Source "->"  
.Destination}}{{end}}'
```

Ne pas afficher `DB_PASSWORD` dans le rapport.

## Taille et fichiers

```
df -hT  
du -sh CHEMIN_MEDIA  
find CHEMIN_MEDIA -type f | wc -l
```

Pour les bibliothèques externes :

```
du -sh CHEMIN_BIBLIOTHEQUE_EXTERNE  
find CHEMIN_BIBLIOTHEQUE_EXTERNE -type f | wc -l
```

## Version de PostgreSQL et extensions

```
docker exec immich_postgres psql -U postgres -d immich -c 'select version();'  
docker exec immich_postgres psql -U postgres -d immich -c '\dx'
```

Adapter conteneur, utilisateur et base.

## Choisir la stratégie de version

Stratégie conservatrice :

1. restaurer avec la même version Immich que la source ;
2. vérifier le contenu ;
3. sauvegarder à nouveau ;
4. effectuer la montée de version.

Stratégie directe : restaurer le dump dans la version cible et laisser Immich appliquer les migrations. Elle est plus rapide mais ajoute des changements simultanés.

Pour une source v2.7.5 et une cible v3.1.0, la stratégie conservatrice facilite le diagnostic.

# Critères de succès

- nombre d'actifs cohérent ;
- albums et partages présents ;
- utilisateurs accessibles ;
- originaux ouvrables ;
- bibliothèques externes scannables ;
- OAuth Web et mobile fonctionnel ;
- jobs sans erreur critique ;
- dump de la nouvelle instance validé.

# 9.2 - Préparer et exporter le serveur source

## Première copie sans interruption

Créer une destination de préparation et copier les médias pendant que la source fonctionne :

```
rsync -aHn --numeric-ids --info=progress2 \  
CHEMIN_MEDIA_SOURCE/ \  
UTILISATEUR@DESTINATION:/chemin/immich-data/
```

Après vérification :

```
rsync -aH --numeric-ids --info=progress2 \  
CHEMIN_MEDIA_SOURCE/ \  
UTILISATEUR@DESTINATION:/chemin/immich-data/
```

Ne pas utiliser `--delete` pour cette première migration.

## Créer un dump de test

```
mkdir -p /srv/export-immich  
  
docker exec -t immich_postgres \  
pg_dump --clean --if-exists --dbname=immich --username=postgres \  
| gzip > /srv/export-immich/immich-test.sql.gz  
  
gzip -t /srv/export-immich/immich-test.sql.gz
```

## Fenêtre de bascule

1. annoncer l'interruption ;
2. terminer les uploads mobiles ;
3. attendre les tâches indispensables ;
4. activer la maintenance ou arrêter le serveur ;
5. créer le dump final ;
6. faire un rsync final ;
7. relever les sommes et tailles ;
8. garder la source intacte.

## Dump final

```
docker exec -it immich_server immich-admin enable-maintenance-mode

docker exec -t immich_postgres \
  pg_dump --clean --if-exists --dbname=immich --username=postgres \
  | gzip > /srv/export-immich/immich-final.sql.gz

gzip -t /srv/export-immich/immich-final.sql.gz
sha256sum /srv/export-immich/immich-final.sql.gz \
  > /srv/export-immich/immich-final.sql.gz.sha256
```

Si la maintenance n'est pas disponible sur la version source, arrêter le service serveur tout en gardant PostgreSQL disponible, ou planifier une courte interruption complète après le dump.

## Copie finale des médias

```
rsync -aH --numeric-ids --info=progress2 \
  CHEMIN_MEDIA_SOURCE/ \
  UTILISATEUR@DESTINATION:/chemin/immich-data/
```

Copier aussi le dump, les fichiers Compose et une version expurgée de l'inventaire. Conserver `.env` dans un canal sécurisé.

## Bibliothèques externes

Deux choix :

- conserver les fichiers au même emplacement réseau et monter le même chemin interne ;
- copier la bibliothèque vers TrueNAS et conserver le même chemin vu par le conteneur.

Le second choix nécessite une vérification des nombres de fichiers et de quelques sommes.

# 9.3 - Transférer et préparer la destination

## Préparer TrueNAS

Exemple :

```
/mnt/HDD_DATA_TRUENAS/IMMICH/data  
/mnt/HDD_DATA_TRUENAS/IMMICH/pgData
```

Créer les datasets et permissions avant de copier. Ne pas démarrer Immich avec des montages incomplets.

## Copier vers le dataset

Depuis la source :

```
rsync -aHn --numeric-ids --info=progress2 \  
CHEMIN_MEDIA_SOURCE/ \  
truenas_admin@ADRESSE_TRUENAS:/mnt/HDD_DATA_TRUENAS/IMMICH/data/
```

Après vérification :

```
rsync -aH --numeric-ids --info=progress2 \  
CHEMIN_MEDIA_SOURCE/ \  
truenas_admin@ADRESSE_TRUENAS:/mnt/HDD_DATA_TRUENAS/IMMICH/data/
```

La connexion SSH peut ne pas autoriser l'écriture directe. Dans ce cas, copier vers un dossier de transit contrôlé puis déplacer depuis TrueNAS avec les droits appropriés.

## Placer le dump

```
sudo mkdir -p /mnt/HDD_DATA_TRUENAS/IMMICH/data/backups
sudo cp immich-final.sql.gz \
    /mnt/HDD_DATA_TRUENAS/IMMICH/data/backups/
```

Corriger les permissions via l'interface TrueNAS ou avec des commandes ciblées après identification des UID.

# Configurer les montages

- données internes vers `/data` ;
- chaque dataset séparé vers `/data/NOM` ;
- bibliothèque externe vers `/mnt/external-libraries/NOM` ;
- mêmes chemins internes que la source si possible.

## Version cible

Si l'application TrueNAS ne propose que la version majeure récente :

- utiliser la restauration onboarding avec le dump compatible ;
- ou valider d'abord le dump dans une instance Docker Compose temporaire à la version source ;
- conserver la source et les sauvegardes jusqu'à réussite.

## Avant démarrage

```
sudo find /mnt/HDD_DATA_TRUENAS/IMMICH/data -maxdepth 1 -mindepth 1 -printf '%f\n'
sudo du -sh /mnt/HDD_DATA_TRUENAS/IMMICH/data
sudo ls -lh /mnt/HDD_DATA_TRUENAS/IMMICH/data/backups
```

Attendre les dossiers :

```
backups
encoded-video
library
profile
thumbs
upload
```

Selon la configuration source, `library` peut être vide ou non utilisé.

# Bascule DNS

Tester d'abord en local. Modifier ensuite Caddy vers `ADRESSE_TRUENAS:30041`, recharger et tester OAuth et mobile. Ne pas changer simultanément domaine, chemins internes et identité OIDC si cela peut être évité.

# 9.4 - Cas pratique Proxmox vers TrueNAS

Ce cas reprend la migration de ton ancienne instance Proxmox vers TrueNAS.

## Situation

- source : VM ou conteneur sur DreamProxmox ;
- destination : TrueNAS Community Edition sur le HPE ML30 Gen9 ;
- dataset : `/mnt/HDD_DATA_TRUENAS/IMMICH` ;
- GPU : Quadro P2000 5 Go ;
- domaine Immich : `immich.dreamsaphir.net` ;
- fournisseur OIDC : `auth.dreamsaphir.net` ;
- port TrueNAS observé : `30041` ;
- ancienne base : PostgreSQL avec pgvecto.rs ;
- méthode déjà utilisée : dump SQL et copie des actifs.

## Procédure recommandée

1. relever la version exacte sur Proxmox ;
2. déclencher un dump PostgreSQL ;
3. copier les six dossiers Immich ;
4. copier les bibliothèques externes ou conserver leur montage ;
5. préparer `data` et `pgData` sur TrueNAS ;
6. configurer l'application et les chemins internes ;
7. restaurer le dump par l'onboarding ou une procédure contrôlée ;
8. vérifier les migrations pgvecto.rs vers VectorChord ;
9. tester localement sur le port 30041 ;
10. basculer Caddy ;
11. tester Authentik et les applications mobiles ;
12. activer la P2000 après validation sur CPU ;
13. déclencher un nouveau dump sur TrueNAS ;
14. mettre en place snapshots et copie vers Proxmox.

# Contrôle des données

Comparer source et destination :

```
find CHEMIN_SOURCE -type f | wc -l  
du -sh CHEMIN_SOURCE  
  
sudo find /mnt/HDD_DATA_TRUENAS/IMMICH/data -type f | wc -l  
sudo du -sh /mnt/HDD_DATA_TRUENAS/IMMICH/data
```

Les nombres peuvent différer si des fichiers temporaires ou générés sont recréés. Comparer surtout les originaux dans `upload`, `library` et la bibliothèque externe.

## Validation fonctionnelle

- compte administrateur ;
- utilisateurs ;
- chronologie ;
- dix photos anciennes et récentes ;
- plusieurs vidéos ;
- albums ;
- favoris ;
- visages et recherche ;
- bibliothèques externes ;
- application mobile ;
- OAuth ;
- sauvegarde de base ;
- accélération GPU.

## Conserver la source

Garder la VM Proxmox arrêtée, non publiée et non modifiée pendant la période de validation. Ne pas lancer les deux instances en écriture derrière le même domaine.

Après validation, archiver la configuration source et conserver au moins un dump et une copie cohérente des médias selon la rétention choisie.