

Immich

En bref

Niveau : avancé

Durée : 45 minutes

Prérequis : Traefik et Authentik en place ; 4 Go de mémoire sur la VM

Niveau de test : reconstitué

Immich, c'est quoi ?

Immich est une galerie de photos et de vidéos auto-hébergée : sauvegarde automatique depuis le téléphone, albums partagés, recherche par visage, par lieu ou par contenu. C'est l'alternative libre aux services de photos en ligne. Il se compose de plusieurs conteneurs qui travaillent ensemble.

Élément	Rôle
Serveur Immich	Interface web, API, envoi et traitement des fichiers
Machine learning	Reconnaissance de visages et recherche par contenu
PostgreSQL (image Immich)	Base de données, avec les extensions de recherche vectorielle
Redis (Valkey)	File des tâches de fond et cache
Fichier <code>immich.json</code>	Configuration déclarative, dont la connexion SSO

Téléphone ou navigateur → Traefik → Serveur Immich → Machine learning, PostgreSQL, Redis

Concrètement, la page suit le même modèle que Nextcloud : un fichier Compose et des labels Traefik. Deux différences. Le SSO est déclaré dans un fichier de configuration lu au démarrage, et non par une commande après coup. Et le premier compte administrateur est créé par le rôle, pour éviter qu'un inconnu s'inscrive le premier.

Prérequis

1. Traefik publie `photos.penhoat.example` (variable `client_subdomains.immich`) et Authentik répond sur `sso.penhoat.example`.
2. `immich` figure dans `nebulouse_apps` et dans `nebulouse_catalogue`.

3. Deux secrets existent dans le Vault du client : `vault_immich_db_password` (déjà prévu) et `vault_immich_admin_password` (à ajouter).
4. La VM dispose d'au moins 4 Go de mémoire : la reconnaissance de visages est gourmande.

`vault_immich_admin_password` complète la liste de la page « Inventaire et variables ». Valeur d'exemple : `FICTIF-admin-motdepasse-6`.

Arborescence du rôle

```
roles/immich/  
├─ defaults/main.yml  
├─ tasks/main.yml  
└─ templates/  
    ├─ docker-compose.yml.j2  
    └─ immich.json.j2
```

Variables

```
# roles/immich/defaults/main.yml  
  
# Version d'Immich : "v2" suit la version majeure 2. Immich publie souvent des changements  
# incompatibles : lire les notes de version avant toute montée de version.  
immich_version: "v2"  
  
# Image PostgreSQL d'Immich. Elle embarque les extensions de recherche vectorielle.  
# Le tag change à chaque version : le reprendre dans le docker-compose.yml officiel  
# de la version d'Immich choisie.  
immich_postgres_image: "ghcr.io/immich-app/postgres:14-vectorchord0.4.3-pgvector0.2.0"  
  
# Dossiers de travail sur la VM.  
immich_dir: /opt/nebuleuse/immich  
# Bibliothèque : photos et vidéos des utilisateurs. C'est le dossier qui grossit.  
immich_library_dir: "{{ immich_dir }}/library"
```

```
# Nom de domaine de l'application.
immich_host: "{{ client_subdomains.immich }}.{{ client_domain }}"

# Secrets : définis dans le Vault, jamais en clair ici.
immich_db_password: "{{ vault_immich_db_password }}"
immich_admin_password: "{{ vault_immich_admin_password }}"

# Réseaux d'où Immich accepte l'en-tête X-Forwarded-For (réseaux privés de Docker).
immich_trusted_proxies: "172.16.0.0/12"

# Réseau Docker partagé avec Traefik.
immich_network: "{{ docker_proxy_network }}"
```

Le fichier Docker Compose

```
# roles/immich/templates/docker-compose.yml.j2
# Fichier géré par Ansible : toute modification manuelle sera écrasée.
# Il contient des secrets : accès réservé à root.

# Variables communes au serveur (ancrage YAML : un seul endroit à modifier).
x-immich-env: &immich_env
  DB_HOSTNAME: database
  DB_USERNAME: immich
  DB_DATABASE_NAME: immich
  DB_PASSWORD: "{{ immich_db_password }}"
  REDIS_HOSTNAME: redis
  # Active la lecture de la configuration déclarative.
  IMMICH_CONFIG_FILE: /config/immich.json
  IMMICH_TRUSTED_PROXIES: "{{ immich_trusted_proxies }}"
  TZ: Europe/Paris

services:
  server:
    image: ghcr.io/immich-app/immich-server:{{ immich_version }}
    container_name: {{ client_name }}_immich_server
    restart: unless-stopped
    environment: *immich_env
    volumes:
```

```

# Bibliothèque des utilisateurs.
- {{ immich_library_dir }}/data
# Configuration déclarative, en lecture seule.
- {{ immich_dir }}/immich.json:/config/immich.json:ro
- /etc/localtime:/etc/localtime:ro

depends_on:
- redis
- database

networks:
- {{ immich_network }}
- internal

labels:
- "traefik.enable=true"
- "traefik.docker.network={{ immich_network }}"
- "traefik.http.routers.immich-{{ client_name }}.rule=Host(`{{ immich_host }}`)"
- "traefik.http.routers.immich-{{ client_name }}.entrypoints=websecure"
# Port d'écoute du serveur Immich.
- "traefik.http.services.immich-{{ client_name }}.loadbalancer.server.port=2283"

```

machine-learning:

```

image: ghcr.io/immich-app/immich-machine-learning:{{ immich_version }}
container_name: {{ client_name }}_immich_ml
restart: unless-stopped

volumes:
# Modèles téléchargés au premier usage : évite de les retélécharger à chaque démarrage.
- {{ immich_dir }}/model-cache:/cache
# Joint uniquement par le serveur, sur le réseau privé.

networks:
- internal

```

redis:

```

image: docker.io/valkey/valkey:8-bookworm
container_name: {{ client_name }}_immich_redis
restart: unless-stopped

networks:
- internal

```

database:

```

image: {{ immich_postgres_image }}
container_name: {{ client_name }}_immich_db

```

```

restart: unless-stopped
environment:
  POSTGRES_USER: immich
  POSTGRES_DB: immich
  POSTGRES_PASSWORD: "{{ immich_db_password }}"
  # Détecte la corruption silencieuse des données.
  POSTGRES_INITDB_ARGS: "--data-checksums"
volumes:
  # La base doit être sur un disque local, jamais sur un partage réseau.
  - {{ immich_dir }}/db-data:/var/lib/postgresql/data
networks:
  - internal

networks:
  {{ immich_network }}:
    external: true
  internal: {}

```

Comme pour Authentik et Nextcloud, seul le serveur est sur le réseau `proxy`. Le service de machine learning, Redis et la base ne sont joignables que depuis le projet.

La configuration déclarative

Immich se configure dans son interface d'administration, mais accepte aussi un fichier JSON lu au démarrage. Ce fichier rend la configuration reproductible pour chaque client. En contrepartie, une fois le fichier actif, les réglages qu'il définit ne sont plus modifiables dans l'interface.

```

# roles/immich/templates/immich.json.j2
{
  "server": {
    "externalDomain": "https://{{ immich_host }}"
  },
  "passwordLogin": {
    "enabled": true
  },
  "oauth": {
    "enabled": true,
    "issuerUrl": "{{ oidc_issuer }}"
  }
}

```

```

"clientId": "{{ oidc_client_id }}",
"clientSecret": "{{ oidc_client_secret }}",
"scope": "openid email profile",
"signingAlgorithm": "RS256",
"buttonText": "Se connecter avec Authentik",
"autoRegister": true,
"autoLaunch": false,
"storageLabelClaim": "preferred_username"
}
}

```

Le format JSON n'accepte pas les commentaires : les réglages sont donc expliqués ci-dessous. Le fichier contient le secret client du SSO, d'où le mode `0600` à son dépôt.

Réglage	Effet
<code>server.externalDomain</code>	Adresse publique, utilisée dans les liens de partage
<code>passwordLogin.enabled</code>	Garde la connexion par mot de passe : le compte administrateur local reste utilisable à côté du SSO
<code>oauth.issuerUrl</code>	Adresse de l'émetteur : Immich y trouve la découverte automatique de la configuration
<code>oauth.signingAlgorithm</code>	Algorithme de signature des jetons attendu (voir les pièges)
<code>oauth.autoRegister</code>	Crée le compte Immich à la première connexion par le SSO
<code>oauth.autoLaunch</code>	Si <code>true</code> , la page de connexion redirige directement vers le SSO
<code>oauth.storageLabelClaim</code>	Champ du jeton utilisé comme étiquette de stockage de l'utilisateur

Déploiement

```

# roles/immich/tasks/main.yml

- name: Créer les dossiers d'Immich
  ansible.builtin.file:
    path: "{{ item }}"
    state: directory
    mode: "0755"
  loop:

```

- "{{ immich_dir }}"
- "{{ immich_library_dir }}"
- "{{ immich_dir }}/model-cache"
- "{{ immich_dir }}/db-data"

La connexion SSO est créée avant le fichier de configuration : elle fournit
l'identifiant et le secret qui y sont inscrits.

- name: Créer la connexion SSO d'Immich dans Authentik

ansible.builtin.include_tasks: ../../authentik/tasks/oidc_app.yml

vars:

oidc_app_name: Immich

oidc_app_slug: immich

oidc_redirect_uris:

Connexion depuis le navigateur.

- matching_mode: strict

url: "https://{{ immich_host }}/auth/login"

Association d'un compte SSO depuis les réglages de l'utilisateur.

- matching_mode: strict

url: "https://{{ immich_host }}/user-settings"

Application mobile.

- matching_mode: strict

url: "app.immich:///oauth-callback"

- name: Déposer la configuration d'Immich

ansible.builtin.template:

src: immich.json.j2

dest: "{{ immich_dir }}/immich.json"

Le fichier contient le secret client : root uniquement.

mode: "0600"

register: immich_config

no_log: true

- name: Déposer le fichier Docker Compose d'Immich

ansible.builtin.template:

src: docker-compose.yml.j2

dest: "{{ immich_dir }}/docker-compose.yml"

mode: "0600"

no_log: true

```

- name: Démarrer Immich
  community.docker.docker_compose_v2:
    project_src: "{{ immich_dir }}"
    state: present

# Immich ne relit son fichier de configuration qu'au démarrage.
- name: Redémarrer Immich si la configuration a changé
  community.docker.docker_compose_v2:
    project_src: "{{ immich_dir }}"
    state: restarted
  when: immich_config.changed

# Le premier démarrage applique les migrations de la base : cela peut durer quelques minutes.
- name: Attendre que l'API d'Immich réponde
  ansible.builtin.command:
    argv:
      - curl
      - -fsS
      - -k
      - --resolve
      - "{{ immich_host }}:443:127.0.0.1"
      - "https://{{ immich_host }}/api/server/ping"
  register: immich_ping
  until: immich_ping.rc == 0
  retries: 60
  delay: 5
  changed_when: false

- name: Savoir si Immich a déjà un administrateur
  ansible.builtin.command:
    argv:
      - curl
      - -sS
      - -k
      - --resolve
      - "{{ immich_host }}:443:127.0.0.1"
      - "https://{{ immich_host }}/api/server/config"
  register: immich_server_config
  changed_when: false

```

```
# Le premier compte créé dans Immich devient administrateur : le rôle le crée
# avant que quiconque puisse s'inscrire.
- name: Créer le compte administrateur du client
  ansible.builtin.command:
    argv: >-
      {{ ['curl', '-sS', '-k', '--fail-with-body',
          '--resolve', immich_host ~ ':443:127.0.0.1',
          '-X', 'POST', '-H', 'Content-Type: application/json',
          '--data-binary', (immich_admin_payload | to_json),
          'https://' ~ immich_host ~ '/api/auth/admin-sign-up'] }}
  vars:
    immich_admin_payload:
      email: "{{ client_admin_email }}"
      password: "{{ immich_admin_password }}"
      name: "Administrateur {{ client_name }}"
  when: not (immich_server_config.stdout | from_json).isInitialized
  no_log: true
```

Vérification

```
# Sur la VM : quatre conteneurs "Up".
docker ps --filter name=immich --format '{{.Names}}\t{{.Status}}'

# L'API répond : doit afficher {"res":"pong"}.
curl -sS https://photos.penhoat.example/api/server/ping

# L'instance est initialisée et le SSO est activé.
curl -sS https://photos.penhoat.example/api/server/config | jq '{isInitialized,
oauthButtonText}'

# La découverte de l'émetteur répond et indique l'algorithme de signature (attendu : RS256).
curl -sS https://sso.penhoat.example/application/o/immich/.well-known/openid-configuration \
| jq '{issuer, id_token_signing_alg_values_supported}'
```

Dans le navigateur, la page <https://photos.penhoat.example> doit afficher le formulaire de connexion avec le bouton « Se connecter avec Authentik ». La connexion SSO avec l'adresse e-mail du client doit ouvrir le compte administrateur créé par le rôle, sans en créer un second.

Pièges

1. **Le premier compte créé devient administrateur** : si Immich est exposé avant la création du compte, la première personne qui s'inscrit, ou qui se connecte par le SSO, prend ce rôle. Le rôle crée donc l'administrateur dès le démarrage.
2. **Réglages verrouillés dans l'interface** : avec le fichier `immich.json`, les rubriques qu'il définit sont en lecture seule. Toute modification passe par le fichier, puis un redémarrage.
3. **Algorithme de signature** : Immich attend des jetons signés en RS256. Authentik ne signe en RS256 que si le provider porte une clé de signature. C'est pourquoi la page Authentik associe la clé de signature par défaut à chaque provider.
4. **Trois adresses de retour** : le navigateur, les réglages utilisateur et l'application mobile en utilisent chacune une. Sans la troisième, la connexion fonctionne dans le navigateur mais échoue sur le téléphone.
5. **Image PostgreSQL** : le tag doit correspondre à la version d'Immich. Une image PostgreSQL ordinaire ne contient pas les extensions de recherche, et le serveur ne démarre pas. Un changement de version majeure de PostgreSQL impose une migration, pas un simple changement de tag.
6. **Base sur un partage réseau** : PostgreSQL y est lent et peut se corrompre. Le dossier `db-data` reste sur un disque local.
7. **Mémoire et premier lancement** : le service de machine learning télécharge ses modèles au premier usage et consomme plusieurs Go. Les premières tâches de reconnaissance sont longues, sans que ce soit une panne.
8. **La bibliothèque grossit vite** : elle croît avec chaque photo et vidéo envoyées. La surveillance de l'espace disque et la sauvegarde du dossier `library` et de la base sont à prévoir, la sauvegarde de la base seule ne suffit pas.
9. **Montées de version** : Immich publie régulièrement des changements incompatibles. Il faut lire les notes de version et sauvegarder avant de changer `immich_version`.
10. **Compte SSO et compte existant** : Immich rattache la connexion SSO au compte qui porte la même adresse e-mail. L'adresse du compte d'Authentik doit donc être identique à celle du compte Immich.
11. **Immich ne joint pas le SSO** : la découverte de l'émetteur échoue depuis le conteneur. Le mécanisme d'alias réseau de Traefik, décrit dans sa page, évite ce détour par l'adresse publique.

Page suivante : **Vaultwarden**.

Revision #1

Created 2026-10-17 13:54:44 UTC by clement-derouet

Updated 2026-10-17 13:54:44 UTC by clement-derouet