

Documentation Docker

L'architecture de chaque cluster est spécifique au pays mais ils suivent un modèle quasi similaire avec les composants suivants :

- **Database (PostgreSQL)** : Stockage des données
- **Backend (Node.js)** : API REST et logique métier
- **Frontend (React + Nginx)** : Interface utilisateur
- **API IA (Python)** : Services d'intelligence artificielle
- **ETL (Python)** : Extraction, transformation et chargement des données

Réseaux :

Chaque pays dispose de son propre réseau :

- USA : Réseau pour les États-Unis
- France : Réseau pour la France
- Suisse : Réseau pour la Suisse

Images Docker

1. Image Backend (Node.js)

Cette image est construite à partir de Node.js 18 Alpine pour une taille réduite. Elle crée un utilisateur non-privilégié 'backend' pour la sécurité, installe uniquement les dépendances de production et nettoie le cache npm pour optimiser l'espace disque.

2. Image Frontend (React + Nginx)

L'image utilise un build multi-étapes : la première étape compile l'application React avec Node.js, puis la seconde étape utilise uniquement Nginx Alpine pour servir les fichiers statiques. Cette approche réduit la taille de l'image et inclue la configuration Nginx.

3. Image API IA (Python)

Basée sur Python 3.11 slim, cette image installe les dépendances système nécessaires pour PostgreSQL (gcc, g++, libpq-dev), crée un utilisateur 'app' non-privilégié et utilise Uvicorn comme serveur ASGI pour optimiser les performances de l'API.

4. Image ETL (Python)

Similaire à l'image API IA mais spécialisée pour les processus ETL. Elle crée un utilisateur dédié 'etl', organise les fichiers dans une structure modulaire et exécute automatiquement le script de migration des données.

Configuration par pays

États-Unis (Cluster complet)

- **Base de données** : oms_usa
- **Locale** : en-US
- **Mode GDPR** : Activé
- **Services** : Tous les composants (PostgreSQL, Backend, Frontend, API IA, ETL)
- **Spécificité** : Gestion de grandes quantités de données

France (Conformité GDPR)

- **Base de données** : oms_france
- **Locale** : fr-FR
- **Mode GDPR** : Strict
- **Services** : PostgreSQL, Backend, Frontend, API IA, ETL
- **Spécificité** : Pas d'API technique (MSPR1)

Suisse (Multi-langues)

- **Base de données** : oms_suisse
- **Langues** : Français, allemand, italien
- **Services** : PostgreSQL, Backend, Frontend, ETL
- **Spécificité** : Pas d'API IA ni de solution Dataviz

Commandes de déploiement par cluster

Cluster États-Unis

```
# Démarrer tous les services USA
docker-compose up -d postgres-usa backend-usa frontend-usa api-ia-usa etl-usa
# Vérifier l'état des services USA
docker-compose ps postgres-usa backend-usa frontend-usa api-ia-usa etl-usa
# Suivre les logs du cluster USA
docker-compose logs -f postgres-usa backend-usa frontend-usa api-ia-usa etl-usa
# Arrêter le cluster USA
docker-compose stop postgres-usa backend-usa frontend-usa api-ia-usa etl-usa
```

Cluster France

Démarrer tous les services France

```
docker-compose up -d postgres-france backend-france frontend-france api-ia-france etl-france
```

Vérifier l'état des services France

```
docker-compose ps postgres-france backend-france frontend-france api-ia-france etl-france
```

Suivre les logs du cluster France

```
docker-compose logs -f postgres-france backend-france frontend-france api-ia-france etl-france
```

Arrêter le cluster France

```
docker-compose stop postgres-france backend-france frontend-france api-ia-france etl-france
```

Cluster Suisse

Démarrer tous les services Suisse

```
docker-compose up -d postgres-suisse backend-suisse frontend-suisse etl-suisse
```

Vérifier l'état des services Suisse

```
docker-compose ps postgres-suisse backend-suisse frontend-suisse etl-suisse
```

Suivre les logs du cluster Suisse

```
docker-compose logs -f postgres-suisse backend-suisse frontend-suisse etl-suisse
```

Arrêter le cluster Suisse

```
docker-compose stop postgres-suisse backend-suisse frontend-suisse etl-suisse
```

Commandes globales

Démarrer tous les clusters

```
docker-compose up -d
```

Construire toutes les images

```
docker-compose build
```

Arrêter tous les services

```
docker-compose down
```

Vérifier l'état de tous les conteneurs

```
docker-compose ps
```

Ports exposés

Service	USA	France	Suisse
PostgreSQL	5432	5433	5434
Backend	4000	4001	4002
Frontend	3000	3002	3004
API IA	8000	8002	-

Accès aux services

- **Frontend USA** : <http://localhost:3000>
- **Frontend France** : <http://localhost:3002>
- **Frontend Suisse** : <http://localhost:3004>
- **API Backend USA** : <http://localhost:4000>
- **API Backend France** : <http://localhost:4001>
- **API Backend Suisse** : <http://localhost:4002>