



MIGRATION DE DONNÉES MÉDICALES VERS MONGODB

Moderniser la gestion des données médicales avec une solution scalable et flexible.

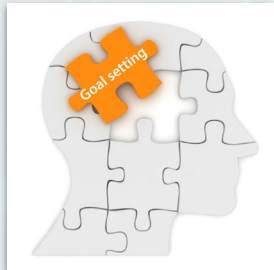
POURQUOI MIGRER LES DONNÉES MÉDICALES ?



Contexte du projet

Le client, une entreprise de santé, gère un volume croissant de données.

Son système actuel est limité et manque de flexibilité.

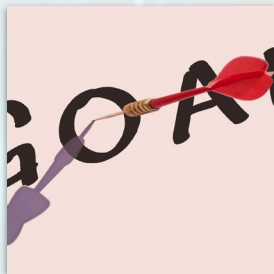


Défis rencontrés

Système obsolète, difficile à faire évoluer.

Traitement inefficace des volumes croissants.

Manque d'intégration avec des technologies modernes.



Objectifs du projet

Migrer les données vers MongoDB pour plus de flexibilité.

Utiliser Docker pour assurer la portabilité.

Préparer un déploiement Cloud évolutif.



UNE MÉTHODOLOGIE EN 4 ÉTAPES



✓ Préparation des données

- ◆ Nettoyage, correction et standardisation pour assurer la qualité.



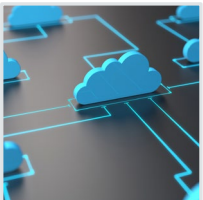
✓ Migration des données vers MongoDB

- ◆ Transfert sécurisé via un script Python optimisé.



✓ Conteneurisation avec Docker

- ◆ Portabilité et déploiement simplifié grâce aux conteneurs.



✓ Exploration Cloud

- ◆ Préparation à une montée en charge évolutive via AWS.





DÉMARCHE TECHNIQUE

1

Nettoyage des données

- ◆ Suppression des doublons et gestion des valeurs manquantes.
- ◆ Standardisation des formats pour assurer la compatibilité avec MongoDB.

2

Scripts Python pour la migration

- ◆ Automatisation du transfert depuis les fichiers CSV.
- ◆ Validation et insertion de plus de **54 000 documents** dans MongoDB.

3

Conteneurisation avec Docker

- ◆ Isolation des services MongoDB et scripts Python.
- ◆ Orchestration avec **Docker Compose** pour un déploiement simplifié.

UNE ARCHITECTURE PENSÉE POUR LA SCALABILITÉ

MongoDB :

- ◆ Stocke les données médicales avec flexibilité et performance.

Scripts Python :

- ◆ Automatisent la migration des données et les tests CRUD.

Conteneurs Docker

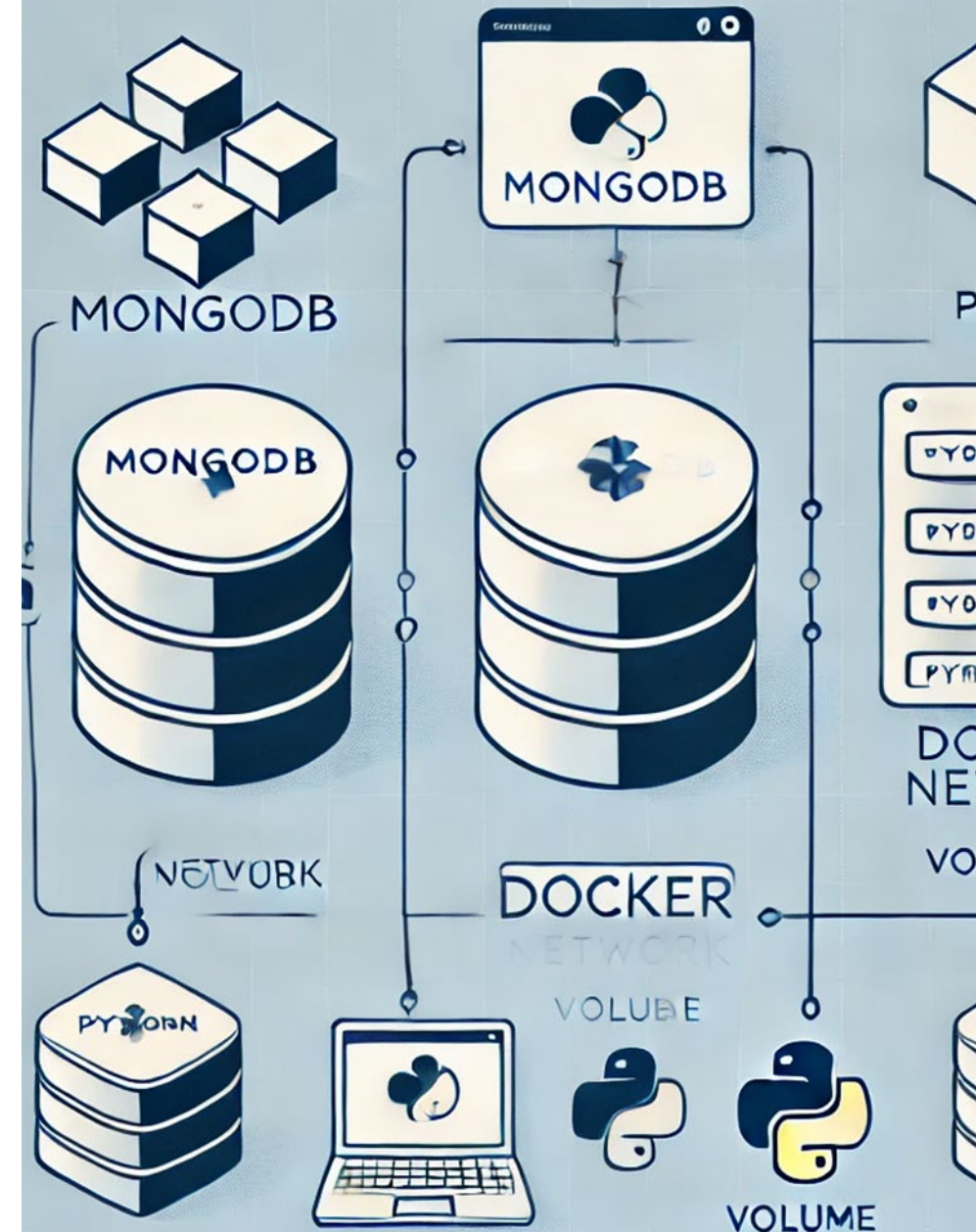
- ◆ Encapsulent les applications et leurs dépendances pour un déploiement portable et efficace.

Volumes Docker

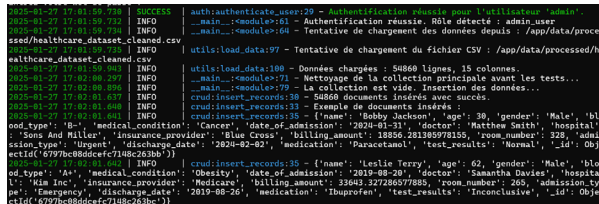
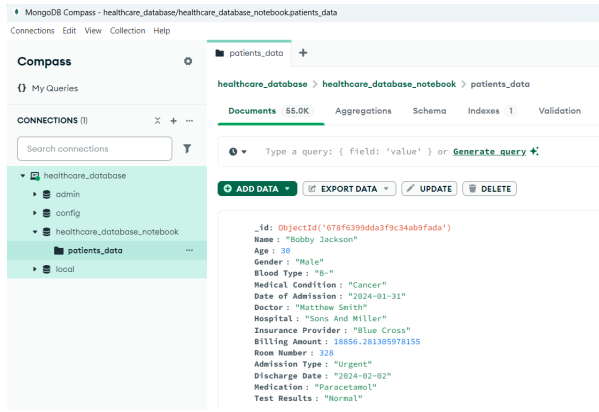
- ◆ Assurent la persistance des données même après redémarrage des conteneurs.

Réseaux Docker

- ◆ Facilitent la communication entre les conteneurs pour une architecture distribuée.



RÉSULTATS CLÉS DE LA MIGRATION



Documents Migrés

✓ 54 860 documents transférés avec succès.

Tests CRUD

✓ Opérations CRUD (Create, Read, Update, Delete) validées à 100 %.

Intégrité des Données

✓ Vérifications complètes : cohérence et exactitude assurées.

Stabilité Docker

✓ Déploiement stable testé dans plusieurs scénarios.

IMPACT DE LA MIGRATION : STABILITÉ ET PERFORMANCE

Vitesse de Migration 🚀

📌 1 000 données transférées par seconde,
assurant un gain de temps considérable.

Tests CRUD Validés ✅

📌 100 % des tests réussis pour garantir
l'intégrité et la cohérence des données.

Infrastructure Moderne 🏗️

- 📌 Scalable, stable et prête pour un déploiement Cloud.
- 📌 Système fiable, adaptable et optimisé pour la gestion des volumes croissants.



DÉMONSTRATION DU PROCESSUS

✓ Démonstration de la migration

Présentation des étapes clés du processus automatisé pour une migration fluide et optimisée.

✓ Utilisation de Docker

Orchestration des conteneurs pour assurer un déploiement portable et stable.

✓ Intégration avec Python

Exécution de scripts automatisés pour gérer les différentes phases de la migration.

✓ Vérification des résultats

Contrôle des données migrées via MongoDB Compass pour garantir leur intégrité.



EXPLORER LES SOLUTIONS AWS

Amazon DocumentDB :

- Base de données managée, compatible MongoDB, idéale pour le stockage et la gestion de volumes massifs de données médicales avec haute disponibilité.

Amazon ECS :

- Orchestration des conteneurs Docker, assurant une gestion efficace et un déploiement scalable des applications.

Amazon S3 :

- Stockage sécurisé, durable et évolutif pour les sauvegardes et l'archivage des données médicales.

COMPARATIF DES SOLUTIONS AWS

Voici un comparatif des solutions AWS pour identifier la plus adaptée aux besoins du projet.

Critère	Amazon DocumentDB	Amazon ECS	Amazon S3
Latence	Faible pour requêtes NoSQL	Dépend du cluster ECS	Très faible en lecture
Coût	Stockage + Requêtes (lecture/écriture)	Basé sur le nombre de conteneurs et les ressources allouées	Stockage économique, mais coût variable selon l'accès aux données.
Scalabilité	Évolutif (ajout d'instances selon la charge)	Haute (ajout dynamique de conteneurs)	Illimité (stockage extensible)
Cas d'usage	Base NoSQL optimisée pour les données médicales.	Déploiement et orchestration de services conteneurisés	Sauvegarde, archivage et stockage d'objets volumineux



JUSTIFICATION DES CHOIX TECHNOLOGIQUES

Choix de MongoDB

- ✓ **Stockage optimisé** des données semi-structurées et volumineuses.
- ✓ **Flexibilité** dans la modélisation et l'évolution des données.

Utilisation de Docker

- ✓ **Isolation et portabilité** des services.
- ✓ **Déploiement rapide** et reproductible dans tous les environnements.

Adoption d'AWS

- ✓ **Scalabilité** à long terme avec DocumentDB.
- ✓ **Gestion efficace** des conteneurs avec ECS.
- ✓ **Stockage sécurisé** et haute disponibilité avec S3.



RECOMMANDATIONS ET PROCHAINES ÉTAPES



Automatisation des tests

Mettre en place des tests automatisés pour garantir l'intégrité et la fiabilité des données après chaque migration. Réduire les risques d'erreur et accélérer le déploiement des mises à jour.



Sauvegardes via Amazon S3

Configurer des sauvegardes régulières sur Amazon S3 pour assurer la sécurité et la disponibilité des données. Garantir un accès rapide et une récupération efficace en cas de besoin.



Développement d'un tableau de bord

Concevoir un tableau de bord interactif avec Power BI ou Tableau pour une analyse avancée. Améliorer l'exploitation des données et faciliter la prise de décision stratégique.

CONCLUSION ET IMPACT DU PROJET

1. Accomplissements clés

- ✓ Migration réussie de 54 860 documents vers MongoDB.
- ✓ Tests CRUD validés à 100 %, garantissant l'intégrité des données.
- ✓ Infrastructure scalable et prête pour une adoption Cloud.

2. Bénéfices pour le client

- 💡 Gestion des données plus flexible et efficace.
- 💡 Un système évolutif, prêt pour des intégrations avec AWS et d'autres services.

3. Vision à long terme

- 🚀 Une base fiable et performante pour intégrer des outils analytiques et améliorer l'aide à la décision.