OCR - PROJET 5

Maintenez et documentez un système de stockage des données sécurisé et performant

Mathilde LE SOLLIEC

31/10/2025

SOMMAIRE

1. Contexte

Problème d'infrastructure de données médicales

Un client nous transmet un dataset de **données médicales de patients.**

Problématique :
Leur système actuel ne suit plus la charge (performance qui chutent, perte de données, augmentation des coûts, risque opérationnel...)

Vient de l'incapacité à gérer:
- **le volume** (quantités de données),
- **et la vélocité** des données médicales (vitesse à laquelle les données dsont produites)

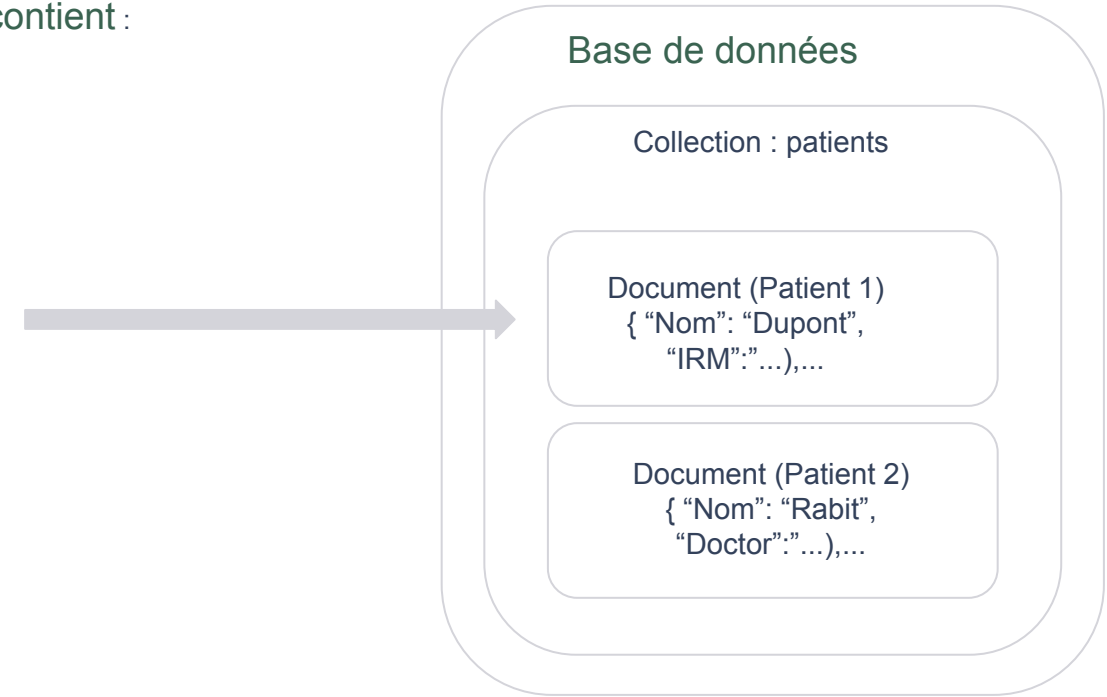
- En tant que data engineer
- Construire une infrastructure de données
- Pour stocker et intégrer facilement des données médicales
- Performant et sécurisé

2.Stockage des données sur MongoDB

Création d'un MongoDB

Chaque document patient contient :

```
{
"Name": "string",
"Age": "int",
"Gender": "string",
"Blood Type": "string",
"Medical Condition": "string",
"Doctor": "string",
"Hospital": "string",
"Room Number": "int",
"Insurance Provider": "string",
"Admission Type": "string",
"Medication": "string",
"Test Results": "string",
"Billing Amount": "float",
"Date of Admission": "datetime",
"Discharge Date": "datetime"
}
```



Création d’un MongoDB

Pourquoi MongoDB

Souplesse du schéma

Pas de modèle fixe, chaque patient peut avoir des suivis très différents. MongoDB permet d'ajouter des champs pour 1 patients

Scalabilité

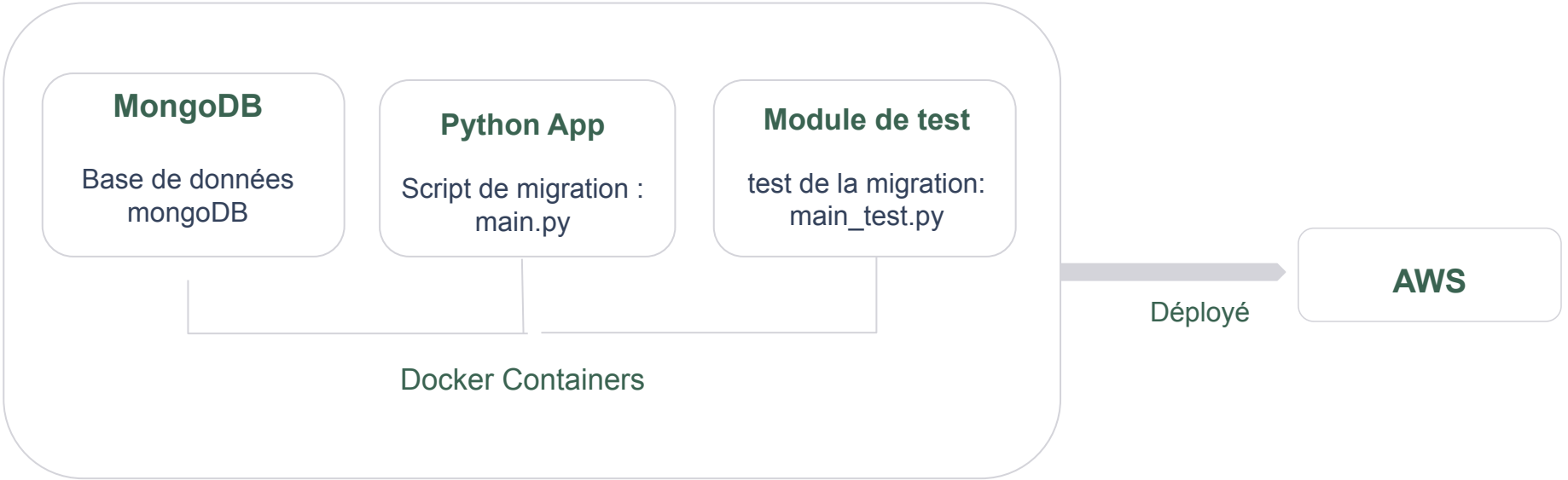
Si l’activité augmente, la base s’agrandit sans interruption

Performance

Requêtes rapide
Accès fluide aux dossiers

3.Architecture

Architecture

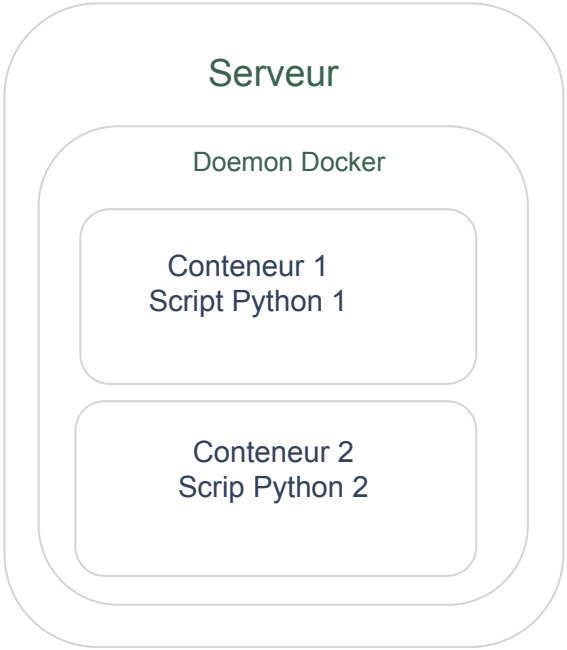


Docker

“emballe” l’ application (code + dépendances + environnement)

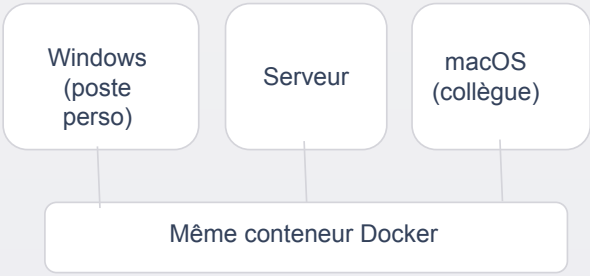
Isolation

Chaque application tourne dans son “contenant”, isolée des autres : pas de conflits



Portabilité

Pas d’erreur de déploiement



AWS

Déployer - faire tourner les conteneurs sur le Cloud (des serveurs distants)

AWS fournit la machine pour exécuter les conteneurs

Sur Amazon ECS : ECS s'occupe de lancer, redémarrer et ajuster les conteneurs selon la charge.

Dans le cadre du projet, les conteneurs ne sont pas réellement déployés sur AWS

Disponibilité et continuité des services

- Moins de cas de panne locale ou d'incident matériel

Scalabilité immédiate

- possibilité d'augmenter facilement la puissance et le stockage.

Services opérationnels

Avantage sauvegarde avec AWS Backup :

- stockage info dans le cloud - reste même si le conteneur est arrêté ou supprimé.
- surveillance (monitoring) - logs de l'application et mongoDB avec CloudWatch

Tarification

Paiement à l'usage (temps, stockage, puissance)
Estimations avec AWS pricing Calculator

4. Démonstration - projet

Contenu du projet - https://githum/Solisdata/ocr_projet5_migration_nosql.git

Fichier	Description
`setup_project.sh`	Script Bash qui automatise l'installation du projet (clone, venv, dépendances, Docker).
`.env`	Variables d'environnement (non suivi par Git) pour sécuriser mots de passe et configurations.
`requirements.txt`	Liste des dépendances nécessaires à l'exécution des scripts.
`data/healthcare_dataset.csv`	Dataset source.
`init-mongo.js`	Script d'initialisation de MongoDB, création des utilisateurs et rôles.
`main.py`	Script principal pour charger, nettoyer et insérer les données dans MongoDB.
`test_main.py`	Tests du script principal.
`docker-compose.yml`	Configuration des conteneurs Docker (MongoDB + application + tests).
`Dockerfile`	Dockerfile pour construire l'image de l'application principale.
`Dockerfile.test`	Dockerfile pour construire l'image du conteneur de tests.

App migration

Charger les données et les insérer dans MongoDB :
docker-compose up -d
docker compose run --rm app python main.py

```
(.venv) PS C:\Users\mathi\OneDrive\Documents\8_OCR\Projet_5\projet_healthcare\ocr_projet5_migration_nosql> d
ocker-compose run --rm app
>>
[+] Creating 1/1
 ✓ Container my_mongo_new  Running                                                    0.0s
Chargement du CSV...
Nettoyage des données...
Connexion à MongoDB...
Insertion des données...
54966 documents insérés.
Script terminé !
(.venv) PS C:\Users\mathi\OneDrive\Documents\8_OCR\Projet_5\projet_healthcare\ocr_projet5_migration_nosql>
```

Tests

3 test :

- `test_clean_dataframe`
- `test_connect_mongo`
- `test_integrity`

docker-compose run app pytest test/main_test.py

```
(.venv) PS C:\Users\mathi\OneDrive\Documents\8_OCR\Projet_5\projet_healthcare\ocr_projet5_migration_nosql> d
ocker-compose run app pytest test/main_test.py
[+] Creating 1/1
 ✓ Container my_mongo_new  Running                                                                    0.0s
========================================= test session starts =========================================
platform linux -- Python 3.13.5, pytest-8.4.2, pluggy-1.6.0
rootdir: /appOCR
collected 3 items

test/main_test.py ...                                                                              [100%]

========================================== 3 passed in 2.69s ==========================================
(.venv) PS C:\Users\mathi\OneDrive\Documents\8_OCR\Projet_5\projet_healthcare\ocr_projet5_migration_nosql>
```

5.Sécurité des données

Sécuriser les données sensibles des patients

- Séparation des rôles et des droits d’accès
- Sécurisation des mots de passe des utilisateurs finaux

Compte technique - MongoDB

3 rôles ont été créés :

- admin : rôle dbAdmin - accès total sur la DB
- writer : rôle readWrite - Lecture/écriture (CRUD)
- reader : rôle read - Lecture Seul

Utilisateurs finaux :

A faire : hachage des mdp

Les mots de passe des utilisateurs finaux ne seront jamais stockés en clair.

Mot de passe saisi → hash bcrypt → comparaison avec hash stocké

Suites - améliorations possibles

- Mettre en place un pipeline CI/CD pour tester automatiquement ton code avec différentes versions de dépendances : Chaque fois que l'on modifies ton code (push sur Git), le pipeline **lance automatiquement des tests**.

- Beaucoup de nouveaux concepts (Docker, AWS, NoSQL) : à pratiquer encore dans les prochains projets

- Possibilité d'utiliser des outils de visualisation comme MongoDB Compass ou Studio 3T