

MAINTENEZ ET DOCUMENTEZ UN SYSTÈME DE STOCKAGE DES DONNÉES SÉCURISÉ ET PERFORMANT

Migration des données médicales vers un système NoSQL, dans un conteneur Docker

Projet Data Engineer – KELLENI Antoine

1

CE QUE NOUS ALLONS DÉCOUVRIR

Déroulé



Contexte



Architecture



Schéma fonctionnel de la base



Script de migration & pipeline loader.py



Orchestration Docker & preuves



Gestion des rôles utilisateurs



Recherche AWS & recommandations



Résultats & bonnes pratiques

ROADMAP DE LA MISSION

- 1. Contexte & Objectifs de la mission**
2. Architecture & Schéma fonctionnel
3. Migration & Pipeline Loader
4. Vérifications MongoDB & Compass
5. Authentification & Sécurité
6. Exploration AWS
7. Résultats & Perspectives

CONTEXTE DE MISSION



- Client en difficulté de scalabilité sur données patients
- Proposition : solution NoSQL scalable avec MongoDB
- Livrables attendus : script d'import, conteneurisation, doc, rôles, AWS

OBJECTIF DE LA MISSION

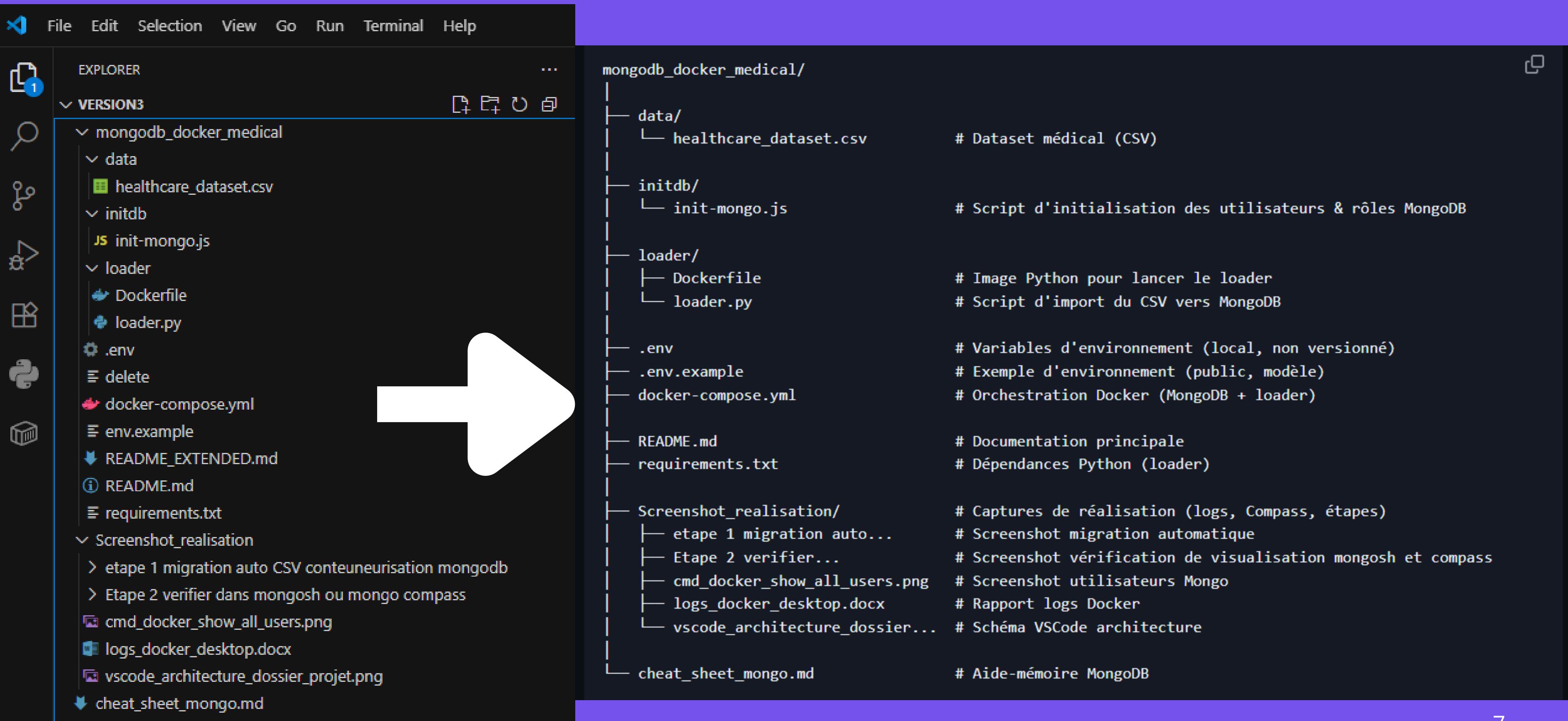


- Migrer un dataset médical CSV vers MongoDB
- Conteneuriser avec Docker
- Créer des rôles utilisateurs sécurisés
- Documenter la démarche technique
- Explorer les options de déploiement AWS

ROADMAP DE LA MISSION

1. Contexte & Objectifs de la mission
- 2. Architecture & Schéma fonctionnel**
3. Migration & Pipeline Loader
4. Vérifications MongoDB & Compass
5. Authentification & Sécurité
6. Exploration AWS
7. Résultats & Perspectives

ARCHITECTURE/STRUCTURE DU PROJET



The image shows a screenshot of the Visual Studio Code (VS Code) interface. The left sidebar displays the project structure under the 'EXPLORER' tab, with a large white arrow pointing from the sidebar towards the code editor area. The code editor on the right contains the following project structure and comments:

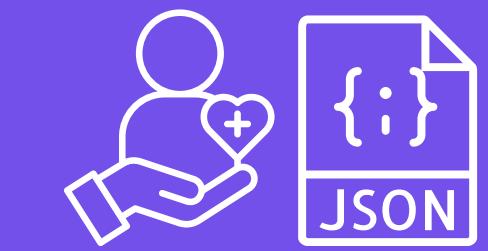
```
mongodb_docker_medical/
├── data/
│   └── healthcare_dataset.csv      # Dataset médical (CSV)
├── initdb/
│   └── init-mongo.js              # Script d'initialisation des utilisateurs & rôles MongoDB
├── loader/
│   ├── Dockerfile                 # Image Python pour lancer le loader
│   └── loader.py                  # Script d'import du CSV vers MongoDB
├── .env                          # Variables d'environnement (local, non versionné)
├── .env.example                  # Exemple d'environnement (public, modèle)
├── docker-compose.yml            # Orchestration Docker (MongoDB + loader)
├── README.md                     # Documentation principale
└── requirements.txt              # Dépendances Python (loader)

Screenshot_realisation/
├── etape 1 migration auto...    # Screenshot migration automatique
├── Etape 2 verifier...          # Screenshot vérification de visualisation mongosh et compass
├── cmd_docker_show_all_users.png # Screenshot utilisateurs Mongo
└── logs_docker_desktop.docx     # Rapport logs Docker

vscode_architecture_dossier...
└── cheat_sheet_mongo.md        # Aide-mémoire MongoDB
```

SCHÉMA FONCTIONNEL DE LA BASE

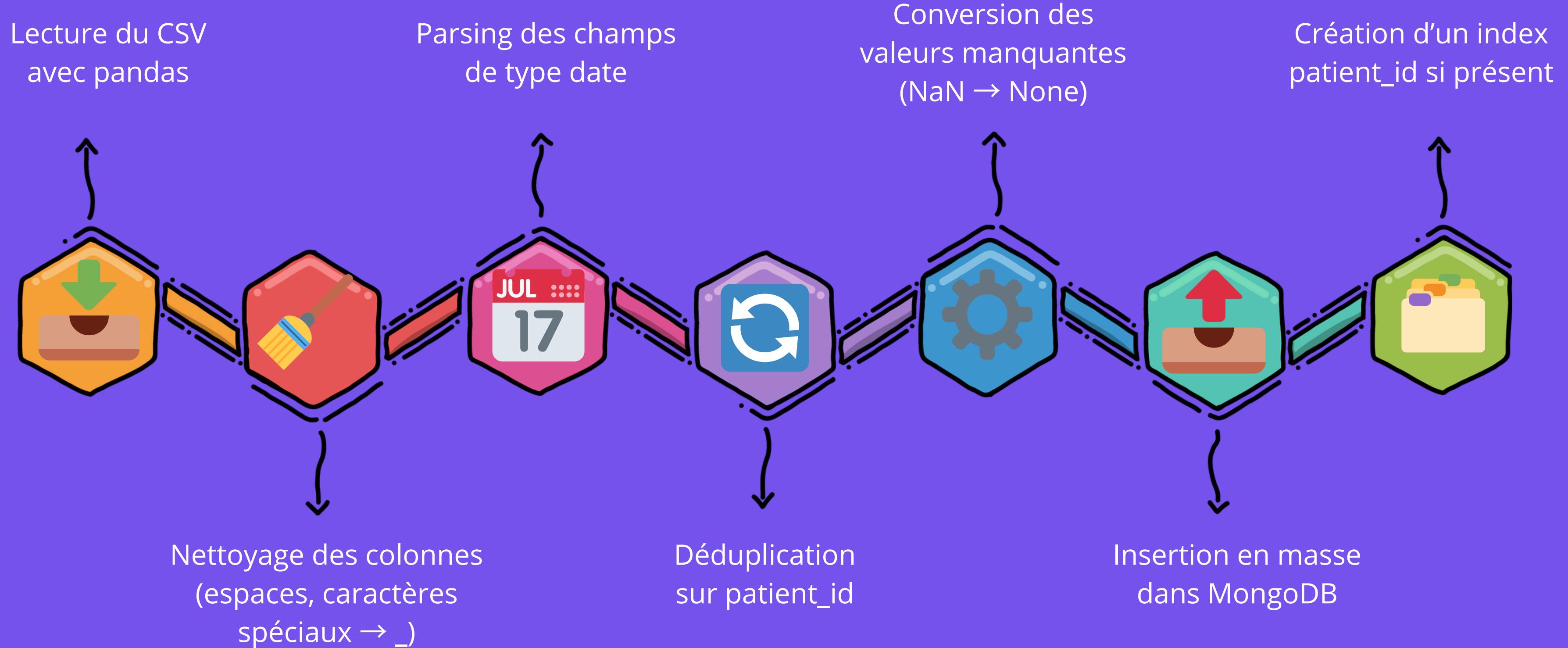
Champ	Type	Description
Name	string	Nom du patient
Age	int	Âge du patient
Gender	string	Genre du patient
Blood_Type	string	Groupe sanguin du patient
Medical_Condition	string	Pathologie principale du patient
Date_of_Admission	date	Date d'admission à l'hôpital
Doctor	string	Médecin en charge du patient
Hospital	string	Nom de l'hôpital où le patient a été admis
Insurance_Provider	string	Assureur du patient
Billing_Amount	float	Montant de la facture associée au patient
Room_Number	int	Numéro de la chambre assignée au patient
Admission_Type	string	Type d'admission (Urgent, Emergency, etc.)
Discharge_Date	date	Date de sortie de l'hôpital
Medication	string	Médication prescrite au patient
Test_Results	string	Résultats des tests médicaux (Normal, Inconclusive, etc.)



ROADMAP DE LA MISSION

1. Contexte & Objectifs de la mission
2. Architecture & Schéma fonctionnel
- 3. Migration & Pipeline Loader**
4. Vérifications MongoDB & Compass
5. Authentification & Sécurité
6. Exploration AWS
7. Résultats & Perspectives

PIPELINE DE MIGRATION (LOADER.PY)



DÉMARRAGE DES CONTENEURS DOCKER



```
PS C:\Users\antoi\OneDrive\Documents\OPENCLASSROOMS\Projet 5 - Maintenez et documentez un système de stockage des données sécurisé et performant\RENDU_PROPRE\Version3\mongodb_docker_medical> docker compose up -d --build
[+] Building 1.4s (12/12) FINISHED
=> [internal] load local bake definitions 0.0s
=> => reading from stdin 733B 0.0s
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 198B 0.0s
=> [internal] load metadata for docker.io/library/python:3.11-slim 1.0s
=> [auth] library/python:pull token for registry-1.docker.io 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [1/4] FROM docker.io/library/python:3.11-slim@sha256:1d6131b5d479888b43200645e03a78443c7157efbdb730e6b48129740727c312 0.0s
=> => resolve docker.io/library/python:3.11-slim@sha256:1d6131b5d479888b43200645e03a78443c7157efbdb730e6b48129740727c312 0.0s
=> [internal] load build context 0.0s
=> => transferring context: 31B 0.0s
=> CACHED [2/4] RUN pip install --no-cache-dir pandas pymongo python-dotenv 0.0s
=> CACHED [3/4] WORKDIR /app 0.0s
=> CACHED [4/4] COPY loader.py /app/loader.py 0.0s
=> exporting to image 0.1s
=> => exporting layers 0.0s
=> => exporting manifest sha256:60b492543473b852eec006c539404d4b5f91f83966cf9faca775a21b7632fb7f 0.0s
=> => exporting config sha256:c7d60efc0da0a3e39092a97d38df7aa8c12776c8d3d60e4eb53d9fd2afb349cf 0.0s
=> => exporting attestation manifest sha256:a8ac9fd5ffa47f14c698b2a9aeb729d19fd6bc519fb4ad61b4eefc70bee43607 0.0s
=> => exporting manifest list sha256:2a476551c95fe6edc6052542f522f3b2454dd1ce99b211077259835e94025c79 0.0s
=> => naming to docker.io/library/mongodb_docker_medical-loader:latest 0.0s
=> => unpacking to docker.io/library/mongodb_docker_medical-loader:latest 0.0s
=> resolving provenance for metadata file 0.0s
[+] Running 5/5
✓ loader 0.0s
✓ Network mongodb_docker_medical_default 0.0s
✓ Volume "mongodb_docker_medical_mongodb_data" 0.0s
✓ Container mongodb 5.9s
✓ Container data_loader 6.0s
```

Build et lancement des conteneurs (MongoDB + Loader) via docker compose up -d --build

STATUT DES SERVICES DOCKER



```
PS C:\Users\antoi\OneDrive\Documents\OPENCLASSROOMS\Projet 5 - Maintenez et documentez un système de stockage des données sécurisé et performant\RENDU_PROPRE\Version3\mongodb_docker_medical> docker compose ps
>>
NAME      IMAGE       COMMAND
mongodb   mongo:7    "docker-entrypoint.s...
          SERVICE     CREATED        STATUS           PORTS
          mongo        15 seconds ago Up 15 seconds (healthy)  0.0.0.0:27017->27017/tcp, [::]:27017->27017/tcp
```

Vérification des conteneurs actifs avec docker compose ps (MongoDB et Loader opérationnels)

LOGS DU LOADER – IMPORT CSV



```
PS C:\Users\antoi\OneDrive\Documents\OPENCLASSROOMS\Projet 5 - Maintenez et documentez un système de stockage des données sécurisé et performant\RENDU_PROPRES\Version3\mongodb_docker_medical> docker compose logs -f loader
>>
data_loader | [loader] Connecting to mongodb://appuser:***@mongodb:27017/healthcare
data_loader | [loader] Reading CSV: /data/healthcare_dataset.csv
data_loader | [loader] Prepared 54966 records (from 55500)
data_loader | [loader] Inserted 54966 documents.
data_loader | [loader] Collection count now: 54966
```

```
PS C:\Users\antoi\OneDrive\Documents\OPENCLASSROOMS\Projet 5 - Maintenez et documentez un système de stockage des données sécurisé et performant\RENDU_PROPRES\Version3\mongodb_docker_medical>
```

The screenshot shows the Docker Desktop interface with the mongo container selected. The logs pane displays the command-line output from the loader container, which includes the connection process, dataset reading, record preparation, document insertion, and the final collection count.

Le loader a bien lu le dataset et inséré 54 966 documents dans MongoDB

ROADMAP DE LA MISSION

1. Contexte & Objectifs de la mission
2. Architecture & Schéma fonctionnel
3. Migration & Pipeline Loader
- 4. Vérifications MongoDB & Compass**
5. Authentification & Sécurité
6. Exploration AWS
7. Résultats & Perspectives

CONNEXION À MONGODB DANS LE CONTENEUR

```
PS C:\Users\antoi\OneDrive\Documents\OPENCLASSROOMS\Projet 5 - Maintenez et documentez un système de stockage des données sécurisé et performant\RENDU_PROPRE\Version3\mongodb_docker_medical> docker exec -it mongodb mongosh -u [REDACTED] -p [REDACTED] --authenticationDatabase admin
>>
Current Mongosh Log ID: 68a2347503a02e2fba89b03c
Connecting to:      mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&authSource=admin&appName=mongosh+2.5.6
Using MongoDB:     7.0.23
Using Mongosh:     2.5.6

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

-----
The server generated these startup warnings when booting
2025-08-17T17:28:17.943+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2025-08-17T17:28:18.530+00:00: vm.max_map_count is too low
-----
```

Connexion réussie au conteneur MongoDB via mongosh avec l'utilisateur root admin

LISTE COMPLÈTE DES UTILISATEURS

Utilisateurs MongoDB – Base admin

```
PS C:\Users\antoi\OneDrive\Documents\OPENCLASSROOMS\Projet 5 - Maintenez et documentez un système de stockage des données sécurisé et performant\RENDU_PROPRES\Version3\mongodb_docker_medical> docker exec -it mongodb mongosh -u admin -p adminpass --authenticationDatabase admin
>>
Current Mongosh Log ID: 68a249c6f5987f942d89b03c
Connecting to:      mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&authSource=admin&appName=mongosh+2.5.6
Using MongoDB:      7.0.23
Using Mongosh:      2.5.6

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

-----
The server generated these startup warnings when booting
2025-08-17T17:28:17.943+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2025-08-17T17:28:18.530+00:00: vm.max_map_count is too low
-----

test> use admin
switched to db admin
admin> db.getUsers()
{
  users: [
    {
      _id: 'admin.admin',
      userId: UUID('9f96db3f-d939-4734-88c5-21ac370adaac'),
      user: 'admin',
      db: 'admin',
      roles: [ { role: 'root', db: 'admin' } ],
      mechanisms: [ 'SCRAM-SHA-1', 'SCRAM-SHA-256' ]
    }
  ],
  ok: 1
}
admin>
```

Vérification des utilisateurs créés dans la base admin

Il est créé automatiquement par l'image officielle MongoDB Docker au premier démarrage.

L'utilisateur admin (root interne Mongo – base admin)

Ses droits sont globaux et illimités : création/suppression de bases, gestion des rôles, configuration, etc.

LISTE COMPLÈTE DES UTILISATEURS

Utilisateurs MongoDB - Base healthcare

```
test> use healthcare
switched to db healthcare
healthcare> show users
[
  {
    _id: 'healthcare.adminUser',
    userId: UUID('466a96b7-784f-484c-bf24-8db997cd8271'),
    user: 'adminUser',
    db: 'healthcare',
    roles: [
      { role: 'dbAdmin', db: 'healthcare' },
      { role: 'clusterAdmin', db: 'admin' },
      { role: 'readWrite', db: 'healthcare' }
    ],
    mechanisms: [ 'SCRAM-SHA-1', 'SCRAM-SHA-256' ]
  },
  {
    _id: 'healthcare.appuser',
    userId: UUID('8deea8cd-3c51-4fef-b92f-05d8a0b77143'),
    user: 'appuser',
    db: 'healthcare',
    roles: [ { role: 'readWrite', db: 'healthcare' } ],
    mechanisms: [ 'SCRAM-SHA-1', 'SCRAM-SHA-256' ]
  },
  {
    _id: 'healthcare.readOnlyUser',
    userId: UUID('8437a920-064a-46be-b2b9-f8a305341834'),
    user: 'readOnlyUser',
    db: 'healthcare',
    roles: [ { role: 'read', db: 'healthcare' } ],
    mechanisms: [ 'SCRAM-SHA-1', 'SCRAM-SHA-256' ]
  },
  {
    _id: 'healthcare.supportUser',
    userId: UUID('e39d6615-f061-4cf7-b6cd-2093edb25348'),
    user: 'supportUser',
    db: 'healthcare',
    roles: [
      { role: 'read', db: 'healthcare' },
      { role: 'dbAdmin', db: 'healthcare' },
      { role: 'readWrite', db: 'healthcare' }
    ],
    mechanisms: [ 'SCRAM-SHA-1', 'SCRAM-SHA-256' ]
  }
]
healthcare>
```

Listing des comptes MongoDB :
(appuser, readOnlyUser, supportUser, adminUser)
crées via init-mongo.js

PREUVES DE MIGRATION (COMPTEUR & DOCUMENT)

```
test> use healthcare
switched to db healthcare
healthcare> db.patients.countDocuments()
54966
```

preuve que le CSV a bien été importé



```
healthcare> db.patients.findOne()
{
  _id: ObjectId('68a211352b906342a2962c42'),
  Name: 'Bobby Jackson',
  Age: 30,
  Gender: 'Male',
  Blood_Type: 'B-',
  Medical_Condition: 'Cancer',
  Date_of_Admission: ISODate('2024-01-31T00:00:00.000Z'),
  Doctor: 'Matthew Smith',
  Hospital: 'Sons and Miller',
  Insurance_Provider: 'Blue Cross',
  Billing_Amount: 18856.281305978155,
  Room_Number: 328,
  Admission_Type: 'Urgent',
  Discharge_Date: ISODate('2024-02-02T00:00:00.000Z'),
  Medication: 'Paracetamol',
  Test_Results: 'Normal'
}
healthcare> db.patients.getIndexes()
[ { v: 2, key: { _id: 1 }, name: '_id_' } ]
```

exemple de document patient

index créés

Vérification que 54 966 documents ont été insérés dans la collection patients et que l'index patient_id a bien été créé.

INDEX DANS MONGO COMPASS

The screenshot shows the MongoDB Compass interface for the 'patients' collection. The 'Indexes' tab is selected, highlighted with a red border. A single index entry for '_id' is listed, also highlighted with a red border. The index is defined as a regular index on the '_id' field. The document count is 55.0K.

Name & Definition	Type	Size	Usage	Properties
_id	REGULAR	561.2 KB	Usage data unavailable	UNIQUE

Par défaut, MongoDB crée un index sur _id.
Cette indexation permet d'optimiser les recherches par _id

```
{  
  "_id": ObjectId('68a211352b906342a2962c42'),  
  "Name": "Bobby Jackson",  
  "Age": 30,  
  "Gender": "Male",  
  "Blood_Type": "B-",  
  "Medical_Condition": "Cancer",  
  "Date_of_Admission": ISODate('2024-01-31T00:00:00.000Z'),  
  "Doctor": "Matthew Smith",  
  "Hospital": "Sons and Miller",  
  "Insurance_Provider": "Blue Cross",  
  "Billing_Amount": 18856.281305978155,
```

DOCUMENTS DE LA COLLECTION PATIENTS



MongoDB Compass - 127.0.0.1:27017/healthcare.patients

Connections Edit View Collection Help

Compass

Welcome patients +

127.0.0.1:27017 > healthcare > patients

Documents 55.0K Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

ADD DATA EXPORT DATA UPDATE DELETE

25 1 - 25 of 54966 Explain Reset Find Options

`_id: ObjectId('68a33161912586c8dbc0352f')`
Name : "Bobby Jackson"
Age : 30
Gender : "Male"
Blood_Type : "B-"
Medical_Condition : "Cancer"
Date_of_Admission : 2024-01-31T00:00:00.000+00:00
Doctor : "Matthew Smith"
Hospital : "Sons and Miller"
Insurance_Provider : "Blue Cross"
Billing_Amount : 18956.281305978155
Room_Number : 328
Admission_Type : "Urgent"
Discharge_Date : 2024-02-02T00:00:00.000+00:00
Medication : "Paracetamol"
Test_Results : "Normal"

`_id: ObjectId('68a33161912586c8dbc03530')`
Name : "Leslie TErRy"
Age : 62
Gender : "Male"
Blood_Type : "A+"Medical_Condition : "Obesity"
Date_of_Admission : 2019-08-20T00:00:00.000+00:00
Doctor : "Samantha Davies"
Hospital : "Kim Inc"
Insurance_Provider : "Medicare"
Billing_Amount : 33643.327286577885
Room_Number : 265
Admission_Type : "Emergency"
Discharge_Date : 2019-08-26T00:00:00.000+00:00
Medication : "Ibuprofen"
Test_Results : "Inconclusive"

Visualisation des documents insérés dans healthcare.patients après la migration CSV → MongoDB



FICHE DE CONNEXION / URI



2017 > healthcare > patients

127.0.0.1:27017

x

Manage your connection settings

- i** While connected, you may only personalize your connection's name, color or favorite status. To fully configure it, you must first disconnect. Beware that disconnecting might cause work in progress to be lost.

Disconnect

URI

```
mongodb://appuser:*****@127.0.0.1:27017/healthcare?  
authSource=healthcare&directConnection=true
```

How do I find my connection string in Atlas?

If you have an Atlas cluster, go to the Cluster view. Click the 'Connect' button for the cluster to which you wish to connect.

[See example](#)

Name

127.0.0.1:27017

Color

No Color

How do I format my connection string?

[See example](#)

Favorite this connection

Favoriting a connection will pin it to the top of your list of connections

Connexion à MongoDB via l'utilisateur applicatif appuser (principe du moindre privilège).

21



docker

VUE TABULAIRE ÉLARGIE



Sessions

Welcome patients +

127.0.0.1:27017 > healthcare > patients

Documents (66.0K) Aggregations Schema Indexes (1) Validation

Type a query: { field: 'value' } or [Generate query +](#)

ADD DATA EXPORT DATA UPDATE DELETE 100 1 - 100 of 54966 Find

* patients

_id	ObjectId	Name String	Age Int32	Gender String	Blood_Type String	Medical_Condition String	Date_of_Admission Date	Doctor String	Hospital String	Insurance_Provider String	Billing_Amount Double	Room_Number Int32	Admission_Type String	Discharge_Date Date	Medication String
1	ObjectId('68a33161912586...')	"Bobby JacksOn"	30	"Male"	"B-"	"Cancer"	2024-01-31T00:00:00.000+...	"Matthew Smith"	"Sons and Miller"	"Blue Cross"	18856.281305978155	328	"Urgent"	2024-02-02T00:00:00.000+...	"Paracetamol"
2	ObjectId('68a33161912586...')	"LesLie TErRy"	62	"Male"	"A+"	"Obesity"	2019-08-20T00:00:00.000+...	"Samantha Davies"	"Kim Inc"	"Medicare"	33643.327286577885	265	"Emergency"	2019-08-26T00:00:00.000+...	"Ibuprofen"
3	ObjectId('68a33161912586...')	"DaNnY sMiTH"	76	"Female"	"A-"	"Obesity"	2022-09-22T00:00:00.000+...	"Tiffany Mitchell"	"Cook PLC"	"Aetna"	27955.096078842456	205	"Emergency"	2022-10-07T00:00:00.000+...	"Aspirin"
4	ObjectId('68a33161912586...')	"andrEw waTtS"	28	"Female"	"O+"	"Diabetes"	2020-11-18T00:00:00.000+...	"Kevin Wells"	"Hernandez Rogers and Va..."	"Medicare"	37909.78240987528	450	"Elective"	2020-12-18T00:00:00.000+...	"Ibuprofen"
5	ObjectId('68a33161912586...')	"adRIENNE bEll"	43	"Female"	"AB+"	"Cancer"	2022-09-19T00:00:00.000+...	"Kathleen Hanna"	"White-White"	"Aetna"	14238.317813937623	458	"Urgent"	2022-10-09T00:00:00.000+...	"Penicillin"
6	ObjectId('68a33161912586...')	"EMILY JOHNSOn"	36	"Male"	"A+"	"Asthma"	2023-12-20T00:00:00.000+...	"Taylor Newton"	"Nunez-Humphrey"	"UnitedHealthcare"	48145.11095104189	389	"Urgent"	2023-12-24T00:00:00.000+...	"Ibuprofen"
7	ObjectId('68a33161912586...')	"edwArD EDWaRDs"	21	"Female"	"AB-"	"Diabetes"	2020-11-03T00:00:00.000+...	"Kelly Olson"	"Group Middleton"	"Medicare"	19580.87234486093	389	"Emergency"	2020-11-15T00:00:00.000+...	"Paracetamol"
8	ObjectId('68a33161912586...')	"ChrIsTiAn MARTinez"	20	"Female"	"A+"	"Cancer"	2021-12-28T00:00:00.000+...	"Suzanne Thomas"	"Powell Robinson and Val..."	"Cigna"	45820.46272159459	277	"Emergency"	2022-01-07T00:00:00.000+...	"Paracetamol"
9	ObjectId('68a33161912586...')	"JASmInE aGuIlA R"	82	"Male"	"AB+"	"Asthma"	2020-07-01T00:00:00.000+...	"Daniel Ferguson"	"Sons Rich and"	"Cigna"	50119.222791548505	316	"Elective"	2020-07-14T00:00:00.000+...	"Aspirin"
10	ObjectId('68a33161912586...')	"ChRISTopher BerG"	58	"Female"	"AB-"	"Cancer"	2021-05-23T00:00:00.000+...	"Heather Day"	"Padilla-Walker"	"UnitedHealthcare"	19784.63106221073	249	"Elective"	2021-06-22T00:00:00.000+...	"Paracetamol"
11	ObjectId('68a33161912586...')	"mIchElle daniELs"	72	"Male"	"O+"	"Cancer"	2020-04-19T00:00:00.000+...	"John Duncan"	"Schaefer-Porter"	"Medicare"	12576.795609050234	394	"Urgent"	2020-04-22T00:00:00.000+...	"Paracetamol"
12	ObjectId('68a33161912586...')	"aaRon MARTiNeZ"	38	"Female"	"A-"	"Hypertension"	2023-08-13T00:00:00.000+...	"Douglas Mayo"	"Lyons-Blair"	"Medicare"	7999.586879604188	288	"Urgent"	2023-09-05T00:00:00.000+...	"Lipitor"
13	ObjectId('68a33161912586...')	"connOR HANSEn"	75	"Female"	"A+"	"Diabetes"	2019-12-12T00:00:00.000+...	"Kenneth Fletcher"	"Powers Miller, and Flor..."	"Cigna"	43282.28335770435	134	"Emergency"	2019-12-28T00:00:00.000+...	"Penicillin"
14	ObjectId('68a33161912586...')	"rObErT bAuer"	68	"Female"	"AB+"	"Asthma"	2020-05-22T00:00:00.000+...	"Theresa Freeman"	"Rivera-Gutierrez"	"UnitedHealthcare"	33207.706633729606	309	"Urgent"	2020-06-19T00:00:00.000+...	"Lipitor"
15	ObjectId('68a33161912586...')	"bROOKE brady"	44	"Female"	"AB+"	"Cancer"	2021-10-08T00:00:00.000+...	"Roberta Stewart"	"Morris-Arellano"	"UnitedHealthcare"	40701.599227308754	182	"Urgent"	2021-10-13T00:00:00.000+...	"Paracetamol"

Vue étendue (100 lignes) pour parcourir rapidement les champs et contrôler la cohérence des types/valeurs. Cette vue me permet de balayer la qualité des données et de repérer rapidement les anomalies éventuelles.

COMPASS SHELL REQUÊTE PAR GENRE



Exécution d'une requête :

```
db.patients.find({ "Gender": "Male" })
```

Retourne tous les patients de sexe masculin.

Cela démontre la capacité de filtrer efficacement les documents dans la collection patients.

The screenshot shows the MongoDB Compass interface with a connection to '127.0.0.1:27017' and a database 'healthcare' containing a 'patients' collection. The query entered is: `> db["patients"].find({ "Gender": "Male" })`. The results pane displays three documents, each representing a patient. The first document is highlighted with a red box around the 'Gender' field. The fields shown for each patient are: _id, Name, Age, Gender, Blood_Type, Medical_Condition, Date_of_Admission, Doctor, Hospital, Insurance_Provider, Billing_Amount, Room_Number, Admission_Type, Discharge_Date, Medication, and Test_Results.

```
_id: ObjectId('68a33161912586c8dbc0352f'),  
Name: 'Bobby JacksOn',  
Age: 30,  
Gender: 'Male',  
Blood_Type: 'B-',  
Medical_Condition: 'Cancer',  
Date_of_Admission: 2024-01-31T00:00:00.000Z,  
Doctor: 'Matthew Smith',  
Hospital: 'Sons and Miller',  
Insurance_Provider: 'Blue Cross',  
Billing_Amount: 18856.281305978155,  
Room_Number: 328,  
Admission_Type: 'Urgent',  
Discharge_Date: 2024-02-02T00:00:00.000Z,  
Medication: 'Paracetamol',  
Test_Results: 'Normal'  
}  
{  
_id: ObjectId('68a33161912586c8dbc03530'),  
Name: 'LesLie TErRy',  
Age: 62,  
Gender: 'Male',  
Blood_Type: 'A+',  
Medical_Condition: 'Obesity',  
Date_of_Admission: 2019-08-20T00:00:00.000Z,  
Doctor: 'Samantha Davies',  
Hospital: 'Kim Inc',  
Insurance_Provider: 'Medicare',  
Billing_Amount: 33643.327286577885,  
Room_Number: 265,  
Admission_Type: 'Emergency',  
Discharge_Date: 2019-08-26T00:00:00.000Z,  
Medication: 'Ibuprofen',  
Test_Results: 'Inconclusive'  
}  
{  
_id: ObjectId('68a33161912586c8dbc03534'),  
Name: 'EMILY JOHNSON',  
Age: 36,  
Gender: 'Male',
```

COMPASS SHELL REQUÊTE PAR ÂGE



Exécution d'une requête :

```
db.patients.find({ "Age": { $gt: 60 } })
```

Retourne uniquement les patients dont l'âge est supérieur à 60 ans.

MongoDB Compass - 127.0.0.1:27017/Shell

Connections Edit View Help

Welcome patients mongosh:127.0.0.1:27017

Compass

My Queries

CONNECTIONS (1)

Search connections

127.0.0.1:27017

healthcare

patients

```
>_MONGOSH
> db["patients"].find({ "Age": { "$gt": 60 } })
)
< [
  {
    "_id": ObjectId('68a33161912586c8dbc03530'),
    "Name": 'LesLie TErRy',
    Age: 62,
    Gender: 'Male',
    Blood_Type: 'A+',
    Medical_Condition: 'Obesity',
    Date_of_Admission: 2019-08-20T00:00:00.000Z,
    Doctor: 'Samantha Davies',
    Hospital: 'Kim Inc',
    Insurance_Provider: 'Medicare',
    Billing_Amount: 33643.327286577885,
    Room_Number: 265,
    Admission_Type: 'Emergency',
    Discharge_Date: 2019-08-26T00:00:00.000Z,
    Medication: 'Ibuprofen',
    Test_Results: 'Inconclusive'
  },
  {
    "_id": ObjectId('68a33161912586c8dbc03531'),
    Name: 'DaNnY sMitH',
    Age: 76,
    Gender: 'Female',
    Blood_Type: 'A-',
    Medical_Condition: 'Obesity',
    Date_of_Admission: 2022-09-22T00:00:00.000Z,
    Doctor: 'Tiffany Mitchell',
    Hospital: 'Cook PLC',
    Insurance_Provider: 'Aetna',
    Billing_Amount: 27955.096078842456,
    Room_Number: 205,
    Admission_Type: 'Emergency',
    Discharge_Date: 2022-10-07T00:00:00.000Z,
    Medication: 'Aspirin',
    Test_Results: 'Normal'
  },
  {
    "_id": ObjectId('68a33161912586c8dbc03537'),
    Name: 'JASmINe aGuIlAr',
    Age: 82,
    Gender: 'Male'
  }
]
```

ROADMAP DE LA MISSION

1. Contexte & Objectifs de la mission
2. Architecture & Schéma fonctionnel
3. Migration & Pipeline Loader
4. Vérifications MongoDB & Compass
- 5. Authentification & Sécurité**
6. Exploration AWS
7. Résultats & Perspectives

AUTHENTIFICATION & RÔLES UTILISATEURS

Les utilisateurs sont créés au démarrage par “**“init-mongo.js”** à partir des variables **.env**.

Utilisateur	Rôles	Usage
admin (root interne) 	Tous les privilèges (root interne MongoDB) sur base admin	Maintenance et opérations privilégiées. À réserver aux admins.
appuser 	readWrite sur DB healthcare	Compte applicatif – principe du moindre privilège.
readOnlyUser 	read sur DB healthcare	Consultation uniquement (reporting, analyse ad hoc).
supportUser 	read, readWrite, dbAdmin sur DB healthcare	Support technique : lecture/écriture et gestion des index & collections.
adminUser 	readWrite, dbAdmin (DB healthcare) + clusterAdmin (DB admin)	Supervision avancée – opérations de haut niveau.

ROADMAP DE LA MISSION

1. Contexte & Objectifs de la mission
2. Architecture & Schéma fonctionnel
3. Migration & Pipeline Loader
4. Vérifications MongoDB & Compass
5. Authentification & Sécurité
- 6. Exploration AWS**
7. Résultats & Perspectives

OPTIONS aws POUR MONGODB



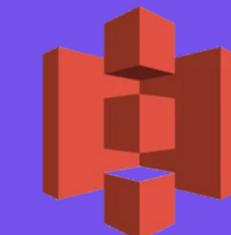
- amazon DocumentDB → service managé, haute dispo, snapshots auto
- Amazon ECS / EC2 → déploiement Docker flexible mais autogéré
- Amazon S3 → sauvegardes régulières (mongodump) haute durabilité
- Amazon CloudWatch → monitoring & alertes centralisées



Amazon
DocumentDB



Amazon ECS



Amazon S3



Amazon Cloudwatch

ROADMAP DE LA MISSION

1. Contexte & Objectifs de la mission
2. Architecture & Schéma fonctionnel
3. Migration & Pipeline Loader
4. Vérifications MongoDB & Compass
5. Authentification & Sécurité
6. Exploration AWS
- 7. Résultats & Perspectives**

RÉSULTATS DU PROJET



Synthèse finale

✓ Migration automatisée → MongoDB

✓ 54 966 documents insérés avec succès

✓ Authentification multi-rôles opérationnelle

✓ Architecture Docker-Compose reproductible

✓ Documentation complète + cheat sheet Mongo

BONNES PRATIQUES & PROCHAINES ÉTAPES

Pourquoi ?

Conclusion

 Principe du moindre privilège pour les comptes applicatifs :

En sécurité informatique, on donne uniquement les droits nécessaires à un utilisateur ou une application, pas plus.

 Créer des index supplémentaires (last_visit_date, {age, gender}) :

Au lieu de scanner tout le dataset, une requête sur "patients femmes de 30 ans" devient quasi instantanée.

 Sauvegardes régulières vers Amazon S3 :

Pour garantir la résilience et éviter toute perte de données (corruption, panne serveur, erreur humaine). Si un conteneur Docker crash et perd les données, on peut restaurer depuis S3 avec mongorestore.

 Monitoring via CloudWatch :

Amazon CloudWatch est l'outil de monitoring centralisé d'AWS. Si le stockage atteint 90 %, CloudWatch peut déclencher une alerte pour agrandir le volume avant que la base ne tombe.

 Roadmap : POC sur ECS → production sur DocumentDB :

Tester l'appli sur ECS avec quelques milliers de documents → si ça marche, migrer vers DocumentDB pour gérer des millions de documents avec sauvegardes et monitoring intégrés

Sécurité (moindre privilège)

Performance (indexation)

Résilience (sauvegardes)

Supervision (monitoring)

Cloud (POC → production)

CONCLUSION



- ✅ Projet validé et opérationnel
- ⚡ Environnement portable & scalable via Docker
- ☁ Préparation réussie au passage cloud (AWS)

MERCI