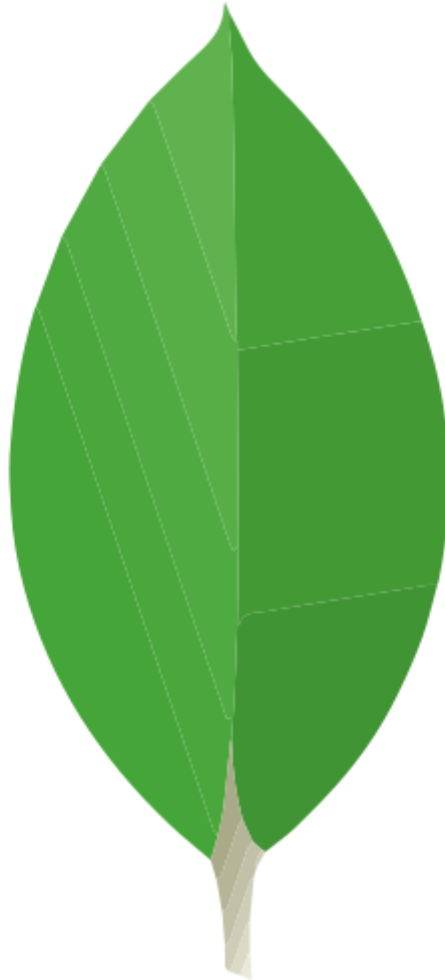


P-NoSQL



Développeur : Thomas Nardou

Client : Mathieu Meylan

Chef de projet : Mathieu Meylan

Périodes : 32

Table des matières


Introduction.....	3
Point A : Restore de la base de données.....	3
Point F : Création des index.....	5
Exemple :	5
Point G : Sauvegarde la base de données	6
Conclusion :	7
Conclusion technique :	7
Conclusion personnelle :	7
ChatGPT :	7

Introduction

Ce projet impliquait la mise en place et l'utilisation d'une base de données MongoDB pour exécuter diverses requêtes. Les tâches comprenaient l'utilisation des opérations "find" et "aggregate", ainsi que la réalisation de recherches textuelles et la gestion des utilisateurs et des rôles. Pour ce faire, un PC de l'ETML était mis à disposition, ainsi qu'un environnement Docker avec un conteneur MongoDB (comprenant le serveur et mongosh), une interface de gestion MongoDB Compass, et VS Code avec l'extension MongoDB. De plus, un accès à Internet était disponible pour les besoins du projet.

Point A : Restore de la base de données

Pouvoir restaurer la base de données il a été nécessaire d'exécuter cette commande dans un environnement **CMD** ou encore **PowerShell**. Voici la commande :



```
docker exec -i mongo mongorestore --uri=mongodb://root:admin@localhost:27017
--authenticationDatabase=admin --gzip --archive=/backupdb/db_mflix.gz
```

Figure 1. Commande pour restaurer la base de données

Cette commande est séparée en deux parties distincts :

- La première partie qui est "**docker exec -i mongo**" qui signifie que l'on va rentrer dans le conteneur Docker qui possède le nom "mongo" et exécuter une commande
- La deuxième partie correspond à la commande qui appartient à mongodb celle-ci permet de restaurer le backup d'une base de données et celle-ci est composée de quatre éléments qui vont être expliquer :
 - o "mongorestore" :
Cette partie signifie que l'on va restaurer le dump d'une base de données
 - o "- -uri" :
Cette partie correspond à la "**connection string**" celle-ci permet de se connecter à un serveur mongodb. Le "mongodb://" indique justement qu'il s'agit d'un serveur mongodb, le "root:admin" qui correspond respectivement l'utilisateur et le mot de passe de l'utilisateur et le "localhost:27017" (ou 127.0.0.1:27017) correspond à l'adresse Ip du serveur suivi du port de connexion (27017)
 - o "- - authenticationDatabase" :
Cette partie permet Spécifier la base de données d'authentification dans laquelle l'utilisateur a été créé dans notre cas il s'agit "d'admin".

- "-gzip" :
Cet élément signifie que pour restaurer la base de données le serveur doit aller chercher de type "gzip" qui sont des fichiers compressés
- "-archive" :
Cette partie permet de restaurer la base de données à partir d'un fichier d'archive dans notre car il s'agit du fichier "db_mflix.gz".

Point F : Création des index

Il serait judicieux de créer un index composite de type text pour les champs "title" et "fullPlot" car ce sont deux éléments qui vont souvent ensemble et que quand un utilisateur fait une recherche il tape les mots clé du film et étant donné que ce sont des valeurs qui vont fréquemment être appelées et aussi parce que ce sont des champs qui ne seront peu voire jamais modifiés. Il serait aussi judicieux de mettre un index text sur le champ "text" de la collection "comments" car c'est aussi un champ qui est appelé souvent pour faire des recherches textuel (si l'on veut chercher un commentaire). Je mettrais aussi un index unique sur le champ "email" de la collection "users" car le courriel peut être utilisé qu'une seule fois et cela permettrait aussi de baisser le temps de recherche lorsqu'un utilisateur se connectera au service.

Exemple :

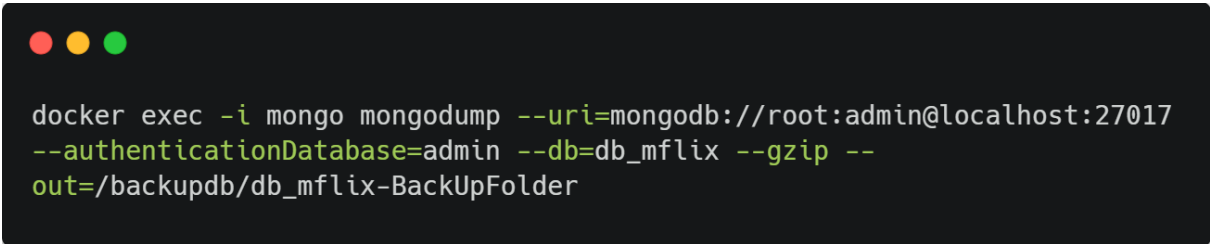
Type	Champ(s)	Requête utilisée	Temps d'exécution sans index [Ms]	Temps d'exécution avec index [Ms]
Composite texte	"title" et "fullplot"	<code>use("db_mflix"); db.movies.find({ \$text: { \$search: "matrix" } });</code>	18	0
texte	"text"	<code>use("db_mflix"); db.comments.find({ \$text: { \$search: "veritatis" } });</code>	21	5
Unique	"email"	<code>use("db_mflix"); db.users.find({email: /\..com\$/});</code>	0	0

Lors de la troisième requête il peut être constater que le temps d'exécution n'a pas changé cela est dû au nombre de document dans la collection (185 au moment de la rédaction du rapport) qui n'est pas assez élevé pour avoir un impacte sur le temps d'exécution de la requête.

Point G : Sauvegarde la base de données

La sauvegarde est un élément très important car elle assure la disponibilité d'une version de la base de données à tout moment. Comme ça en cas d'incident elle permet de restaurer une version antérieure, limitant ainsi les dommages potentiels.

Pour pouvoir sauvegarder la base de données en prenant le moins d'espace de stockage possible, il est nécessaire d'exécuter cette commande dans un environnement **CMD** ou encore **PowerShell** :



```
docker exec -i mongo mongodump --uri=mongodb://root:admin@localhost:27017
--authenticationDatabase=admin --db=db_mflix --gzip --
out=/backupdb/db_mflix-BackUpFolder
```

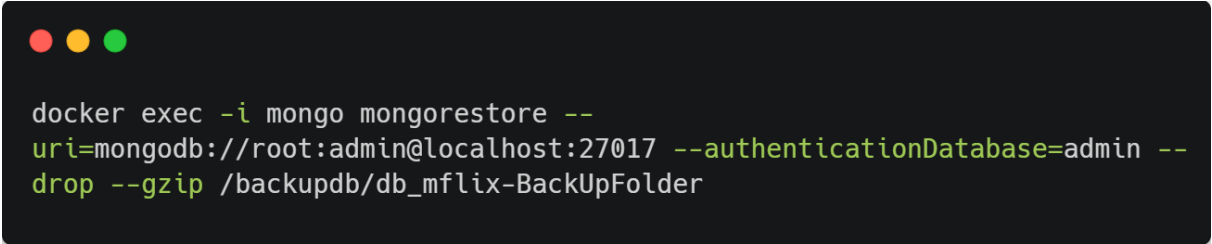
Figure 2. Commande pour sauvegarder sa Base de données

Pour ce qui est de la commande elle ne change pas énormément à la commande pour restaurer la base de données ([Point A](#)) mais il y a quelque point qui changent et qui vont être expliqués :

- mongodump :
Cet élément signifie que l'on va faire un dump/sauvegarde de la base de données.
- --db :
Cet élément veut tout simplement dire que l'on va sauvegarder la base de données qui possède le nom que l'on a rentré dans notre cas il s'agit de la base de données "db_mflix".
- --gzip :
Cela signifie que l'on va compresser le résultat de la commande dans notre cas ce seront les fichiers JSON et BSON.
- --out :
Cette partie signifie l'endroit où la sauvegarde de la base de données se trouvera dans le cas présent la sauvegarde se trouvera dans le dossier "db_mflix-BackUpFolder".

Une fois que l'exécution de la commande s'est terminée un dossier a été créé à l'emplacement spécifié.

Pour pouvoir restaurer la base de données il est nécessaire d'exécuter cette commande toujours dans un environnement **CMD** ou encore **PowerShell** :



```
docker exec -i mongo mongorestore --  
uri=mongodb://root:admin@localhost:27017 --authenticationDatabase=admin --  
drop --gzip /backupdb/db_mflix-BackUpFolder
```

Figure 3. Commande de restauration de la base de données

Cette commande diffère un peu de celle qui est au [Point A](#) car au lieu de spécifier l'endroit où se trouve l'archive (avec le `--archive`) il est nécessaire de spécifier le nom du dossier qui a été créé lors de la sauvegarde. Le "`--drop`" permet quant à lui de supprimer la base de données s'il existe déjà.

Conclusion :

Conclusion technique :

Toutes les requêtes qui ont été demandées dans le cahier des charges ont pu être réalisées avec succès cependant la requête d'agrégation numéro 12 n'est pas complète car s'il y a plusieurs genres qui apparaissent le même nombre de fois alors il y en a qu'un seul qui est retourné.

Conclusion personnelle :

J'ai bien aimé faire ce projet malgré les difficultés que j'ai eu pour faire certaines requêtes notamment les agrégations. Si ce projet était à refaire je le referais avec plaisir mais je changerais certaines choses notamment mon autonomie car je demande souvent de l'aide pour les requêtes difficiles, je ferais aussi plus attention au journal de travail car à certain moment j'oubliais de le mettre à jour.

ChatGPT :

J'ai utilisé l'intelligence artificielle "ChatGPT" afin de reformuler des phrases dans le rapport.