RÉPUBLIQUE DU SÉNÉGAL

ÉCOLE SUPÉRIEURE POLYTECHNIQUE

DÉPARTEMENT GÉNIE INFORMATIQUE

PROJET DE BASES DE DONNEES AVANCEES

La réplication d'une base de données MongoDB sous Windows

Présenté par :

Papa Matar Ndiaye

Cheikh Mbacké Ba

Mouhamadou Moustapha Sy

Khady Diagne

Professeur :

Dr. Mbacké

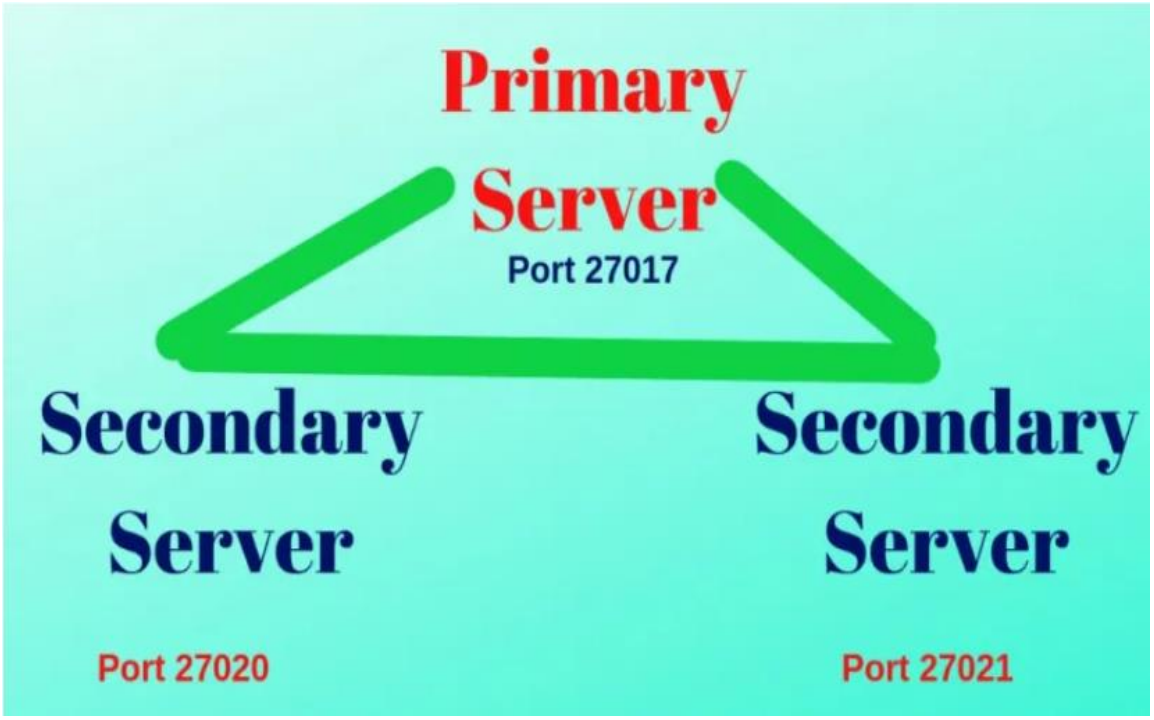
Année universitaire **2022-2023**

Configuration étape par étape de la réplication et de la distribution d'une base de données MongoDB sous Windows

Qu'est-ce que la réplication ?

La réplication est un moyen de conserver des copies identiques des données sur plusieurs serveurs et elle est recommandée pour tous les déploiements de production.

Nous allons essayer d'adopter cette architecture logicielle :



Serveur primaire : Rendre autonome en tant que serveur primaire sans : 27017

Serveurs secondaires : Démarrage de deux serveurs avec les numéros de port : 27020 et 27021 (je configure la réplication sur une seule machine). Si je décide de configurer la réplication sur 3 hôtes différents, je dois indiquer le nom de l'hôte et le numéro de port lorsque cela est requis dans les étapes suivantes.

Configuration étape par étape de la réplication MongoDB sous Windows

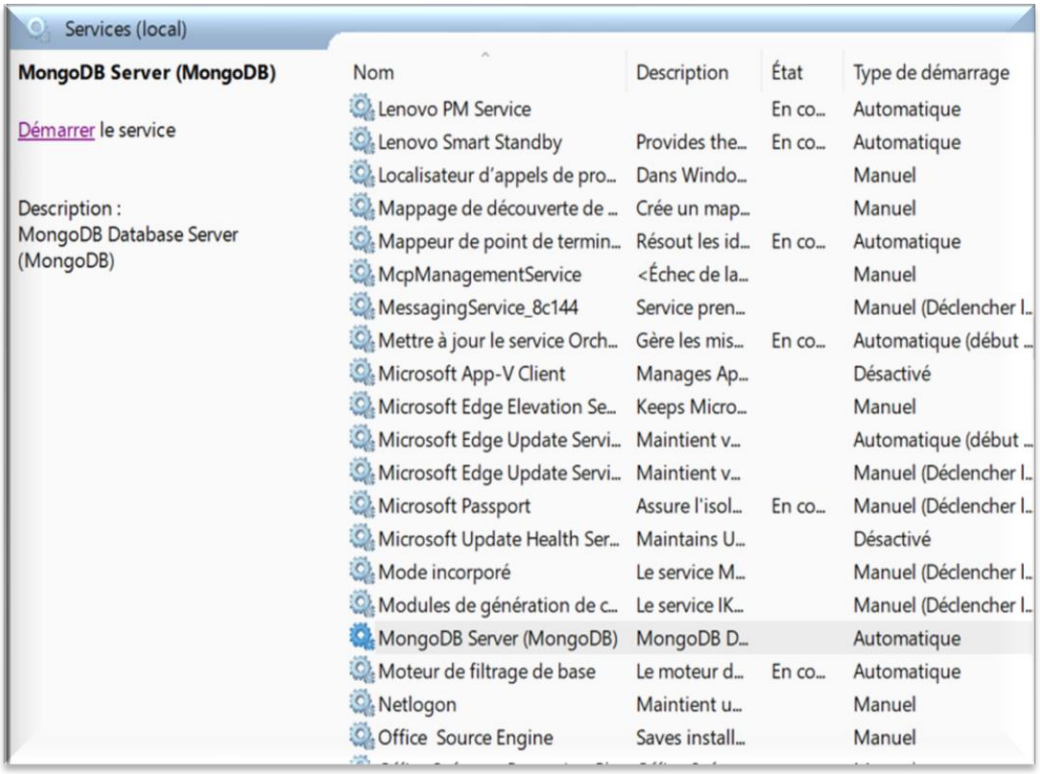
Remarques importantes : Avant de configurer la réplication MongoDB, nous allons effectuer une sauvegarde importante.

Etape 1 : Démarrez le serveur autonome comme indiqué ci-dessous.

```
mongod --dbpath "C:\Program Files\MongoDB\Server\4.0\data" --logpath "C:\Program Files\MongoDB\Server\4.0\log\mongod.log" --port 27017 --storageEngine=wiredTiger --journal --replSet ProjectEsp
```

Ouvrir d'abord le cmd ensuite mettre le chemin d'accès du répertoire d'installation de MongoDB en faisant comme suit : `cd C:\Program Files\MongoDB\Server\4.0\bin` puis ensuite démarrer le serveur.

Remarque : Pour éviter les confusions nous allons d'abord éteindre les services de MongoDB sous Windows et démarrer notre serveur à l'étape 1



Démarrez le serveur autonome comme indiqué ci-dessous.

```
C:\WINDOWS\system32>mongod --dbpath "C:\Program Files\MongoDB\Server\4.0\data" --logpath "C:\Program Files\MongoDB\Serve
r\4.0\log\mongod.log" --port 27017 --storageEngine=wiredTiger --journal --replSet ProjectEsp
2023-03-11T09:54:54.010+0100 I CONTROL  [main] log file "C:\Program Files\MongoDB\Server\4.0\log\mongod.log" exists; mov
ed to "C:\Program Files\MongoDB\Server\4.0\log\mongod.log.2023-03-11T08-54-54".
```

<u>Etape 2</u> : Connectez-vous au serveur avec le numéro de port 27017 avec la commande ci-dessous mongo --port 27017

```
C:\Users\Moustapha
λ mongo --port 27017
MongoDB shell version v4.0.28
connecting to: mongodb://127.0.0.1:27017/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("3d6f1bc3-0dac-4db7-b3ac-ba178d97f619") }
MongoDB server version: 4.0.28
Server has startup warnings:
2023-03-11T09:54:54.652+0100 I CONTROL  [initandlisten]
2023-03-11T09:54:54.652+0100 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2023-03-11T09:54:54.652+0100 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2023-03-11T09:54:54.653+0100 I CONTROL  [initandlisten]
2023-03-11T09:54:54.653+0100 I CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2023-03-11T09:54:54.653+0100 I CONTROL  [initandlisten] **          Remote systems will be unable to connect to this server.
2023-03-11T09:54:54.653+0100 I CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to specify which IP
2023-03-11T09:54:54.653+0100 I CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all to
2023-03-11T09:54:54.653+0100 I CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2023-03-11T09:54:54.653+0100 I CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2023-03-11T09:54:54.653+0100 I CONTROL  [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
```

Etape 3 : Ensuite, créez la variable rsconf avec la commande ci-dessous.

rsconf={_id:"ProjectEsp",members :[{_id:0,host:"localhost:27017"}]}

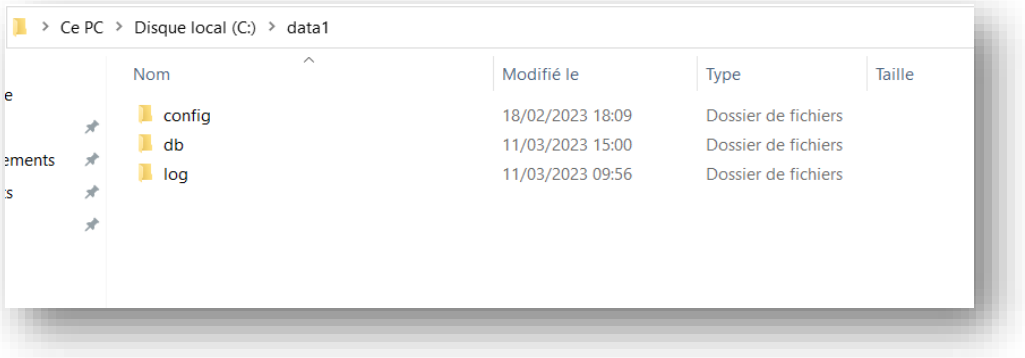
rs.initiate(rsconf)

```
> rsconf={_id:"ProjectEsp",members :[{_id:0,host:"localhost:27017"}]}
{
        "_id" : "ProjectEsp",
        "members" : [
                {
                        "_id" : 0,
                        "host" : "localhost:27017"
                }
        ]
}
> rs.initiate(rsconf)
{ "ok" : 1 }
ProjectEsp:SECONDARY>
ProjectEsp:PRIMARY> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
world   0.000GB
ProjectEsp:PRIMARY>
```

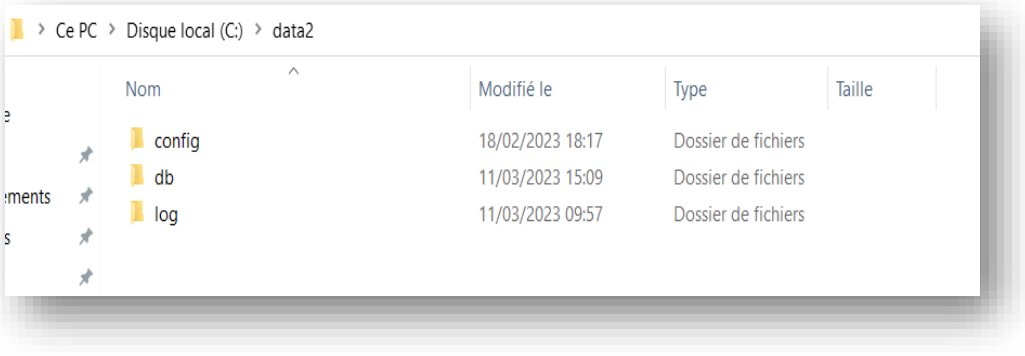
Remarque : Ici, je configure la réplication sur une seule machine Windows.

Etape 4 : Création de deux répertoires data 1 et data 2 pour les ports respectivement 27020 et 27021

Répertoire data1 qui contient le port 27020 :



Répertoire data2 qui contient le port 27021 :



Etape 5 : Démarrez le serveur secondaire sur le port 27020 avec la commande ci-dessous.

mongod --dbpath "C:\data1\db" --logpath "C:\data1\log\mongod.log" --port 27020 --storageEngine=wiredTiger --journal --replSet ProjectEsp

Démonstration :

```
C:\Users\Moustapha
λ mongod --dbpath "C:\data1\db" --logpath "C:\data1\log\mongod.log" --port 27020 --storageEngine=wiredTiger --journal --replSet ProjectEsp
2023-03-11T09:56:57.207+0100 I CONTROL  [main] log file "C:\data1\log\mongod.log" exists; moved to "C:\data1\log\mongod.log.2023-03-11T08-56-57".
```

Etape 6 : Connectez-vous au serveur secondaire avec le port de 27020 la commande ci-dessous.

```
C:\Users\Moustapha
λ mongo --port 27020
MongoDB shell version v4.0.28
connecting to: mongodb://127.0.0.1:27020/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("3b587a2e-10a6-4d88-9002-21573a172484") }
MongoDB server version: 4.0.28
Server has startup warnings:
2023-03-11T09:56:57.854+0100 I CONTROL  [initandlisten]
2023-03-11T09:56:57.854+0100 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2023-03-11T09:56:57.854+0100 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2023-03-11T09:56:57.855+0100 I CONTROL  [initandlisten]
2023-03-11T09:56:57.855+0100 I CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2023-03-11T09:56:57.855+0100 I CONTROL  [initandlisten] **          Remote systems will be unable to connect to this server.
2023-03-11T09:56:57.855+0100 I CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to specify which IP
2023-03-11T09:56:57.855+0100 I CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all to
2023-03-11T09:56:57.855+0100 I CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2023-03-11T09:56:57.855+0100 I CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2023-03-11T09:56:57.855+0100 I CONTROL  [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
```

Etape 7 : Démarrez le serveur secondaire sur le port 27021 avec la commande ci-dessous.
mongod --dbpath "C:\data2\db" --logpath "C:\data2\log\mongod.log" --port 27021 --storageEngine=wiredTiger --journal --replSet ProjectEsp

Démonstration :

```
C:\Users\Moustapha
λ mongod --dbpath "C:\data2\db" --logpath "C:\data2\log\mongod.log" --port 27021 --storageEngine=wiredTiger --journal --replSet ProjectEsp
2023-03-11T09:57:15.239+0100 I CONTROL  [main] log file "C:\data2\log\mongod.log" exists; moved to "C:\data2\log\mongod.log.2023-03-11T08-57-15".
```

Etape 8 : Connectez-vous au serveur secondaire avec le port de 27021 la commande ci-dessous.

```
C:\Users\Moustapha
λ mongo --port 27021
MongoDB shell version v4.0.28
connecting to: mongodb://127.0.0.1:27021/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("a58b3777-652e-486c-b276-6e80c0fc9b5d") }
MongoDB server version: 4.0.28
Server has startup warnings:
2023-03-11T09:57:15.851+0100 I CONTROL  [initandlisten]
2023-03-11T09:57:15.851+0100 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2023-03-11T09:57:15.851+0100 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2023-03-11T09:57:15.851+0100 I CONTROL  [initandlisten]
2023-03-11T09:57:15.852+0100 I CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2023-03-11T09:57:15.852+0100 I CONTROL  [initandlisten] **          Remote systems will be unable to connect to this server.
2023-03-11T09:57:15.852+0100 I CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to specify which IP
2023-03-11T09:57:15.852+0100 I CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all to
2023-03-11T09:57:15.852+0100 I CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2023-03-11T09:57:15.852+0100 I CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2023-03-11T09:57:15.852+0100 I CONTROL  [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
```

Etape 9 : Exécutez les commandes suivantes sur le serveur principal du port 27017.
rs.add("localhost:27020")
rs.add("localhost:27021")

Démonstration :
Pour le rs.add("localhost:27020")

```
ProjectEsp:PRIMARY> rs.add("localhost:27020")
{
        "ok" : 1,
        "operationTime" : Timestamp(1676748841, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1676748841, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
```

Pour le rs.add("localhost:27021")

```
ProjectEsp:PRIMARY> rs.add("localhost:27021")
{
        "ok" : 1,
        "operationTime" : Timestamp(1676749213, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1676749213, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
```

Etape 10 : Allez maintenant sur les serveurs secondaires et exécutez la commande ci-dessous sur les deux serveurs secondaires.
rs.slaveOk()

Démonstration sur les deux serveurs secondaires 27020 et 27021 :

Pour le serveur secondaire avec le port 27020

```
C:\Users\Moustapha
λ mongo --port 27020
MongoDB shell version v4.0.28
connecting to: mongodb://127.0.0.1:27020/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("b96387eb-aad1-41d1-b598-e268f9882138") }
MongoDB server version: 4.0.28
Server has startup warnings:
2023-03-11T15:56:22.901+0100 I CONTROL  [initandlisten]
2023-03-11T15:56:22.901+0100 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2023-03-11T15:56:22.902+0100 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2023-03-11T15:56:22.902+0100 I CONTROL  [initandlisten]
2023-03-11T15:56:22.902+0100 I CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2023-03-11T15:56:22.902+0100 I CONTROL  [initandlisten] **          Remote systems will be unable to connect to this server.
2023-03-11T15:56:22.903+0100 I CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to specify which IP
2023-03-11T15:56:22.903+0100 I CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all to
2023-03-11T15:56:22.903+0100 I CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2023-03-11T15:56:22.903+0100 I CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2023-03-11T15:56:22.903+0100 I CONTROL  [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---

ProjectEsp:SECONDARY> rs.slaveOk()
WARNING: slaveOk() is deprecated and may be removed in the next major release. Please use secondaryOk() instead.
ProjectEsp:SECONDARY> show dbs
admin   0.000GB
config  0.000GB
local   0.001GB
world   0.000GB
ProjectEsp:SECONDARY>
```

Pour le serveur secondaire avec le port 27021

```
C:\Users\Moustapha
λ mongo --port 27021
MongoDB shell version v4.0.28
connecting to: mongodb://127.0.0.1:27021/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("85cd9fc2-de88-4d31-bf39-f1049ffd7d86") }
MongoDB server version: 4.0.28
Server has startup warnings:
2023-03-11T16:07:41.795+0100 I CONTROL  [initandlisten]
2023-03-11T16:07:41.795+0100 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2023-03-11T16:07:41.795+0100 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2023-03-11T16:07:41.796+0100 I CONTROL  [initandlisten]
2023-03-11T16:07:41.796+0100 I CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2023-03-11T16:07:41.796+0100 I CONTROL  [initandlisten] **          Remote systems will be unable to connect to this server.
2023-03-11T16:07:41.796+0100 I CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to specify which IP
2023-03-11T16:07:41.796+0100 I CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all to
2023-03-11T16:07:41.796+0100 I CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2023-03-11T16:07:41.796+0100 I CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2023-03-11T16:07:41.796+0100 I CONTROL  [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---

ProjectEsp:SECONDARY> rs.slaveOk()
WARNING: slaveOk() is deprecated and may be removed in the next major release. Please use secondaryOk() instead.
ProjectEsp:SECONDARY> show dbs
admin   0.000GB
config  0.000GB
local   0.001GB
world   0.000GB
ProjectEsp:SECONDARY>
```

Pour le serveur Principale avec le port 27017 avec la commande ci-dessous :
rs.isMaster()

```
ProjectEsp:PRIMARY> rs.isMaster()
{
        "hosts" : [
                "localhost:27017",
                "localhost:27020",
                "localhost:27021"
        ],
        "setName" : "ProjectEsp",
        "setVersion" : 3,
        "ismaster" : true,
        "secondary" : false,
        "primary" : "localhost:27021",
        "me" : "localhost:27021",
        "electionId" : ObjectId("7fffffff0000000000000002"),
        "lastWrite" : {
                "opTime" : {
                        "ts" : Timestamp(1676751143, 1),
                        "t" : NumberLong(2)
                },
                "lastWriteDate" : ISODate("2023-02-18T20:12:23Z"),
                "majorityOpTime" : {
```

Vérification de la configuration de la réplication :

Créons un serveur principal de collecte et vérifiez que cette modification se répercutera ou non sur les serveurs secondaires.
Apres avoir importer les donnes de world nous allons suivre les étapes suivantes :

Etape 1 : Connectez-vous au serveur principal en utilisant la commande.

```
use world
```

Etape 2 : Créer une collection dans le serveur Primaire après importer les données de la base world

```
db.country.insert({name:"MongoDB"})
```

Etape 3 : Connectez-vous maintenant aux serveurs secondaires et vérifiez la liste de la base de données en exécutant la commande show dbs.

Pour le serveur secondaire avec le port 27020

```
ProjectEsp:SECONDARY> show dbs
admin   0.000GB
config  0.000GB
local   0.001GB
world   0.000GB
ProjectEsp:SECONDARY>
```

Pour le serveur secondaire avec le port 27021

```
ProjectEsp:SECONDARY> show dbs
admin   0.000GB
config  0.000GB
local   0.001GB
world   0.000GB
ProjectEsp:SECONDARY>
```

Etape 4 : Basculez vers la base de données nouvellement créée en utilisant la commande use world.

Pour le serveur secondaire avec le port 27020

```
ProjectEsp:SECONDARY> use world
switched to db world
ProjectEsp:SECONDARY>
```

Pour le serveur secondaire avec le port 27021

```
ProjectEsp:SECONDARY> use world
switched to db world
ProjectEsp:SECONDARY>
```

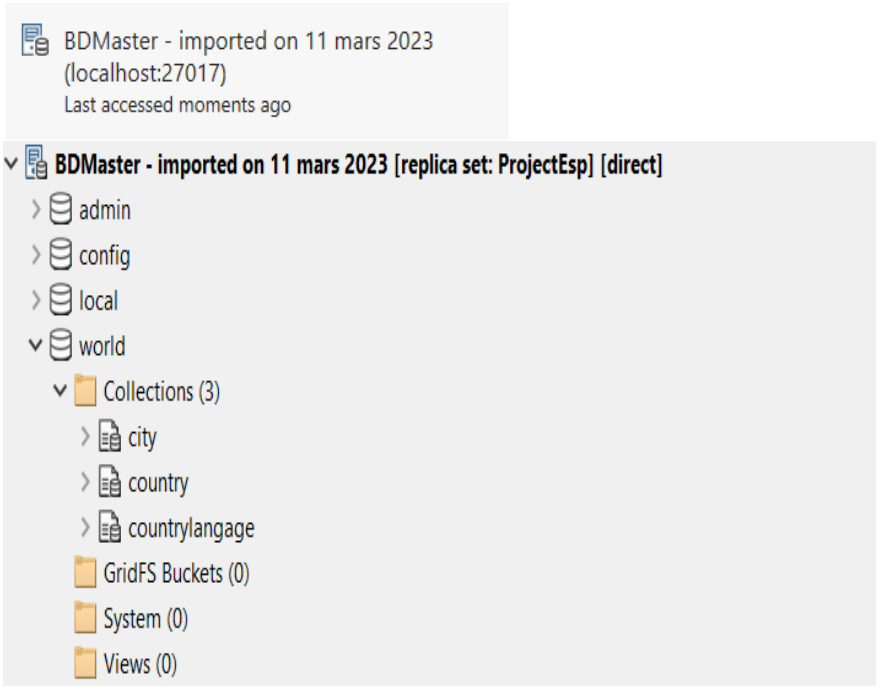
Etape 5 : exécutez la commande sur la base de données world.

```
db.country.find().pretty().
```

Après vérification de configuration de la réplication nous avons conclu que les données ont été bien répliqué et on ne peut que créer des bases de données que dans le serveur primaire et non dans les deux serveurs secondaires. Nous allons le démontrer dans l'arborescence ci-dessous :

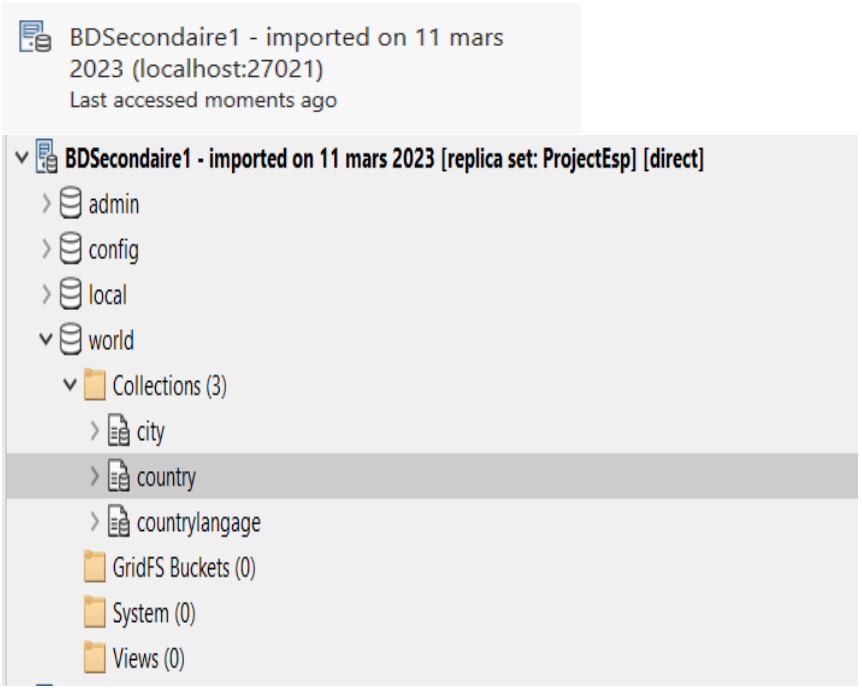
Montrons l'arborescence de notre base de données avec leur port respectif

Pour le serveur Principale avec le port 27017



Pour le serveur secondaire avec le port 27020

Pour le serveur secondaire avec le port 27021

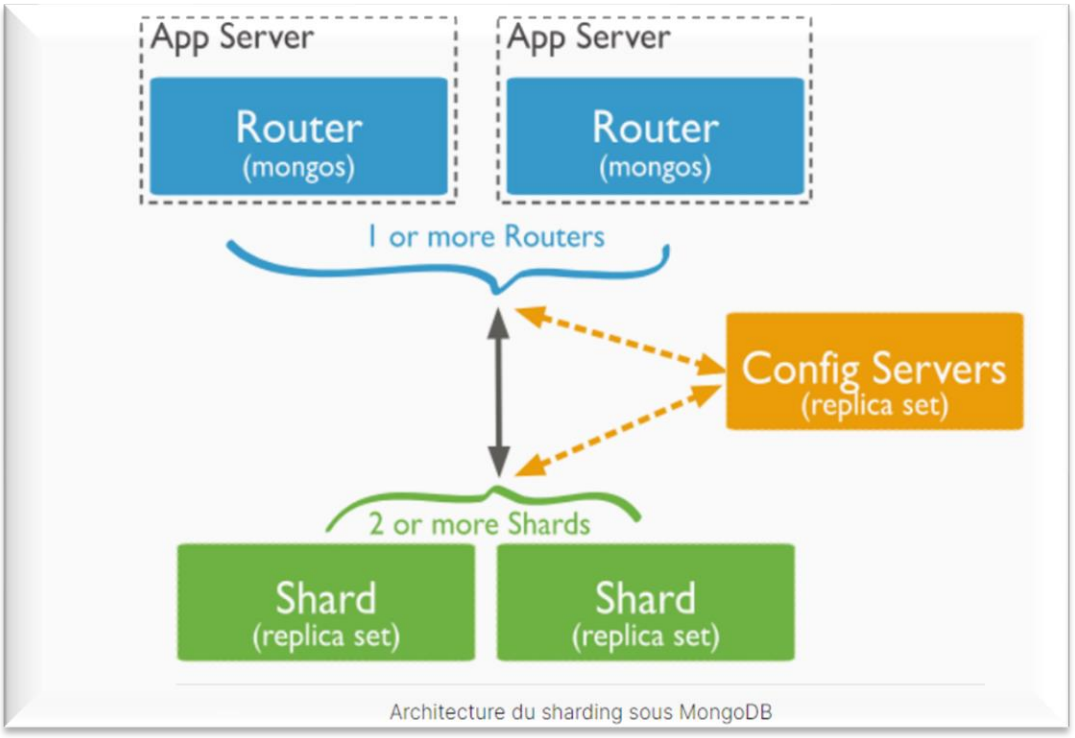


Ainsi, dans cet article, nous avons expliqué comment configurer la réplication MongoDB sous Windows, puis vérifié l'état de la réplication.

Création des shards

Les **shards** contiennent l'ensemble des données, les chunks. Ils peuvent contenir plusieurs chunks, mais pas forcément contigus (tri des données de l'arbre).

L'architecture de distribution d'un cluster est schématisée dans la figure ci-dessous :

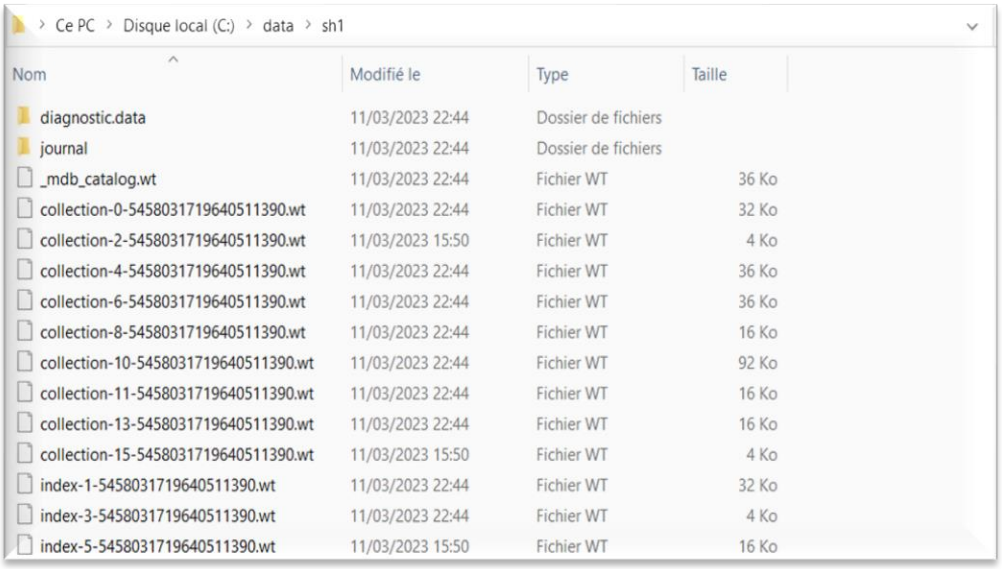


Architecture du sharding sous MongoDB

Ensuite, il est nécessaire de lancer les shard en ReplicaSet. Nous allons en créer deux pour tester la distribution. Pour chaque shard, le paramètre --shardsvr est nécessaire pour permettre son intégration.

D'abord nous allons créer trois répertoires de configuration sh1, sh2 et sh3 qui ont les ports respectifs 27030, 27031 et 27032 pour contenir les shardings pour permettre la distribution de la base de données.

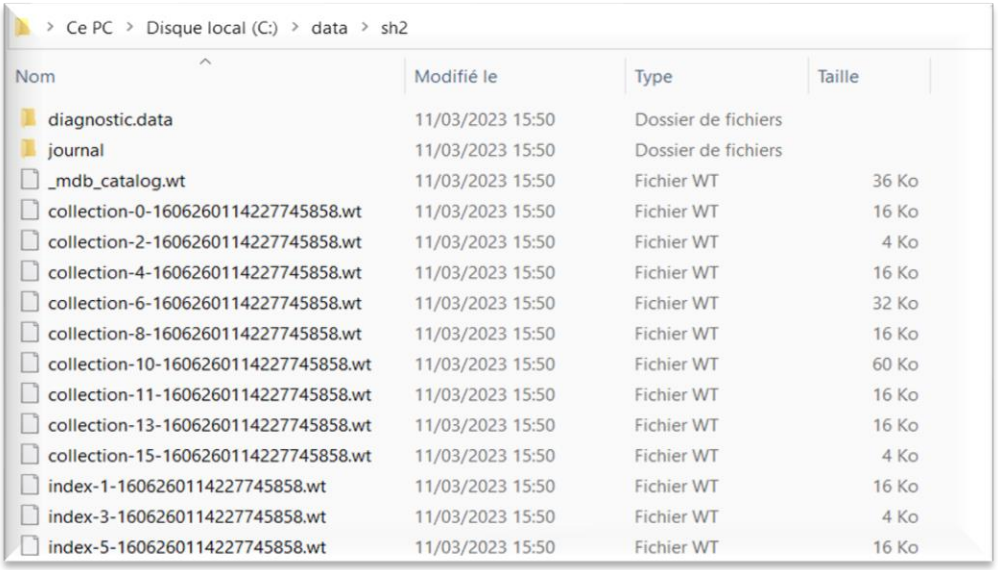
- Pour le premier répertoire de configuration sh1 qui écoute le port 27030



Ce PC › Disque local (C:) › data › sh1

Nom	Modifié le	Type	Taille
diagnostic.data	11/03/2023 22:44	Dossier de fichiers	
journal	11/03/2023 22:44	Dossier de fichiers	
_mdb_catalog.wt	11/03/2023 22:44	Fichier WT	36 Ko
collection-0-5458031719640511390.wt	11/03/2023 22:44	Fichier WT	32 Ko
collection-2-5458031719640511390.wt	11/03/2023 15:50	Fichier WT	4 Ko
collection-4-5458031719640511390.wt	11/03/2023 22:44	Fichier WT	36 Ko
collection-6-5458031719640511390.wt	11/03/2023 22:44	Fichier WT	36 Ko
collection-8-5458031719640511390.wt	11/03/2023 22:44	Fichier WT	16 Ko
collection-10-5458031719640511390.wt	11/03/2023 22:44	Fichier WT	92 Ko
collection-11-5458031719640511390.wt	11/03/2023 22:44	Fichier WT	16 Ko
collection-13-5458031719640511390.wt	11/03/2023 22:44	Fichier WT	16 Ko
collection-15-5458031719640511390.wt	11/03/2023 15:50	Fichier WT	4 Ko
index-1-5458031719640511390.wt	11/03/2023 22:44	Fichier WT	32 Ko
index-3-5458031719640511390.wt	11/03/2023 22:44	Fichier WT	4 Ko
index-5-5458031719640511390.wt	11/03/2023 15:50	Fichier WT	16 Ko

- Pour le deuxième répertoire de configuration sh2 qui écoute le port 27031



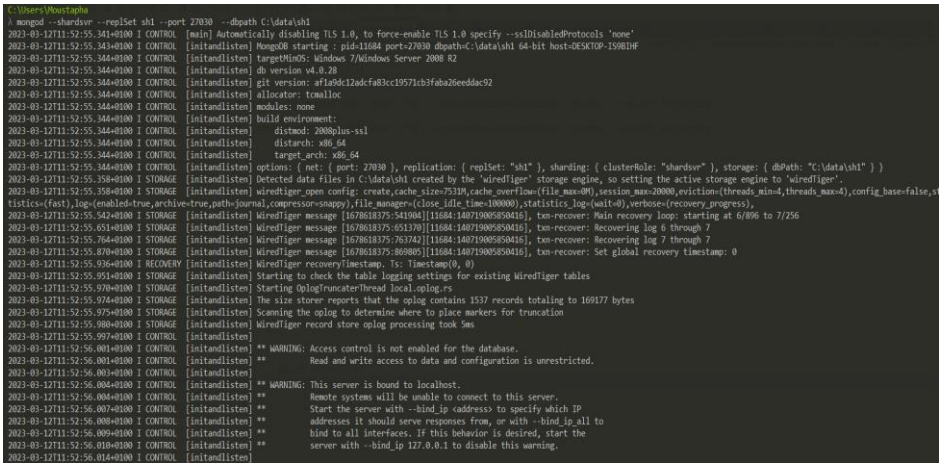
Ce PC › Disque local (C:) › data › sh2

Nom	Modifié le	Type	Taille
diagnostic.data	11/03/2023 15:50	Dossier de fichiers	
journal	11/03/2023 15:50	Dossier de fichiers	
_mdb_catalog.wt	11/03/2023 15:50	Fichier WT	36 Ko
collection-0-1606260114227745858.wt	11/03/2023 15:50	Fichier WT	16 Ko
collection-2-1606260114227745858.wt	11/03/2023 15:50	Fichier WT	4 Ko
collection-4-1606260114227745858.wt	11/03/2023 15:50	Fichier WT	16 Ko
collection-6-1606260114227745858.wt	11/03/2023 15:50	Fichier WT	32 Ko
collection-8-1606260114227745858.wt	11/03/2023 15:50	Fichier WT	16 Ko
collection-10-1606260114227745858.wt	11/03/2023 15:50	Fichier WT	60 Ko
collection-11-1606260114227745858.wt	11/03/2023 15:50	Fichier WT	16 Ko
collection-13-1606260114227745858.wt	11/03/2023 15:50	Fichier WT	16 Ko
collection-15-1606260114227745858.wt	11/03/2023 15:50	Fichier WT	4 Ko
index-1-1606260114227745858.wt	11/03/2023 15:50	Fichier WT	16 Ko
index-3-1606260114227745858.wt	11/03/2023 15:50	Fichier WT	4 Ko
index-5-1606260114227745858.wt	11/03/2023 15:50	Fichier WT	16 Ko

Etape 1 : Démarrez le serveur sharding sur le port 27030 avec la commande ci-dessous.

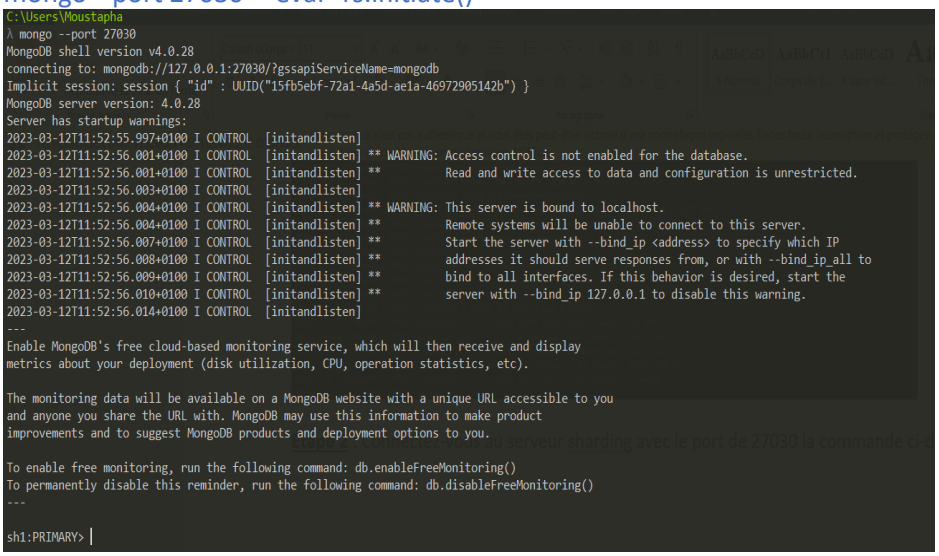
```
mongod --shardsvr --replSet sh1 --port 27030  --dbpath C:\data\sh1
```

Démonstration :



Etape 2 : Connectez-vous au serveur sharding avec le port de 27030 la commande ci-dessous.

mongo --port 27030 --eval "rs.initiate()"

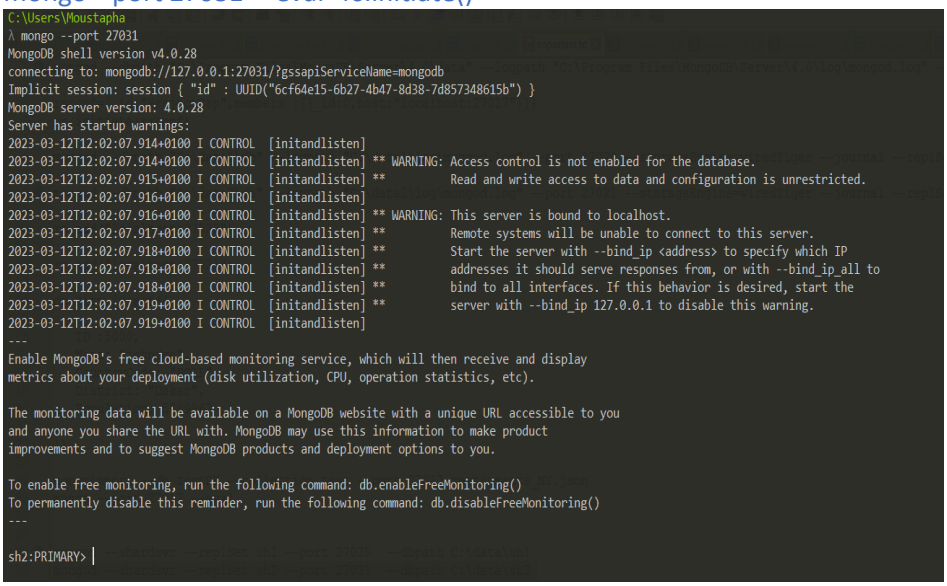


Etape 3 : Démarrez le serveur sharding sur le port 27030 avec la commande ci-dessous.

mongod --shardsvr --replSet sh2 --port 27031 --dbpath C:\data\sh2

Etape 4 : Connectez-vous au serveur sharding avec le port de 27030 la commande ci-dessous.

mongo --port 27031 --eval "rs.initiate()"



L'option **--eval** permet de faire passer la commande **rs.initiate()** directement au serveur puis de récupérer la main sur la console. Ici, cette commande permet d'initialiser le *ReplicaSet* de chaque *shard*.

Etape 5 : Démarrez le serveur sharding sur le port 27032 avec la commande ci-dessous.

mongod --shardsvr --replSet sh3 --port 27032 --dbpath C:\data\sh3

Démonstration :

```
C:\Users\Moustapha
λ mongod --shardsvr --replSet sh3 --port 27032  --dbpath C:\data\sh3
2023-03-14T18:19:40.922+0100 I CONTROL  [main] Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'
2023-03-14T18:19:40.925+0100 I CONTROL  [initandlisten] MongoDB starting : pid=12548 port=27032 dbpath=C:\data\sh3 64-bit host=DESKTOP-IS9BIHF
2023-03-14T18:19:40.926+0100 I CONTROL  [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2023-03-14T18:19:40.926+0100 I CONTROL  [initandlisten] db version v4.0.28
2023-03-14T18:19:40.926+0100 I CONTROL  [initandlisten] git version: af1a9dc12adcfa83cc19571cb3faba26eeddac92
2023-03-14T18:19:40.926+0100 I CONTROL  [initandlisten] allocator: tcmalloc
2023-03-14T18:19:40.926+0100 I CONTROL  [initandlisten] modules: none
2023-03-14T18:19:40.926+0100 I CONTROL  [initandlisten] build environment:
2023-03-14T18:19:40.926+0100 I CONTROL  [initandlisten]     distmod: 2008plus-ssl
2023-03-14T18:19:40.926+0100 I CONTROL  [initandlisten]     distarch: x86_64
```

Etape 6 : Connectez-vous au serveur sharding avec le port de 27032 la commande ci-dessous.
mongo --port 27032

```
C:\Users\Moustapha
λ mongo --port 27032
MongoDB shell version v4.0.28
connecting to: mongodb://127.0.0.1:27032/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("fb708cda-e5bd-4a9a-9b62-3a48b0bce811") }
MongoDB server version: 4.0.28
Server has startup warnings:
2023-03-14T18:19:41.062+0100 I CONTROL  [initandlisten]
2023-03-14T18:19:41.063+0100 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2023-03-14T18:19:41.065+0100 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2023-03-14T18:19:41.069+0100 I CONTROL  [initandlisten]
2023-03-14T18:19:41.070+0100 I CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2023-03-14T18:19:41.071+0100 I CONTROL  [initandlisten] **          Remote systems will be unable to connect to this server.
2023-03-14T18:19:41.071+0100 I CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to specify which IP
2023-03-14T18:19:41.073+0100 I CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all to
2023-03-14T18:19:41.073+0100 I CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2023-03-14T18:19:41.076+0100 I CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2023-03-14T18:19:41.077+0100 I CONTROL  [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---

> rs.initiate()
{
        "info2" : "no configuration specified. Using a default configuration for the set",
        "me" : "localhost:27032",
        "ok" : 1
}
sh3:SECONDARY>
```

Etape 7 : Lancement du mongos et connexion des shards

On interagit avec un cluster de fragment en se connectant a une instance de mongos (Query router) Maintenant que nous avons des shards et des ConfigServers, nous pouvons nous attaquer au mongos (routeur) en utilisant la commande ci-dessous.

```
mongos --configdb "ProjectEsp/localhost:27017" --port 27018
```

```
C:\WINDOWS\system32>mongos --configdb "ProjectEsp/localhost:27017" --port 27018
2023-03-14T10:49:38.019+0100 W SHARDING [main] Running a sharded cluster with fewer than 3 config servers should only be done for testing purposes and is not recommended for production.
2023-03-14T10:49:38.141+0100 I CONTROL  [main] Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'
2023-03-14T10:49:38.145+0100 I CONTROL  [main]
2023-03-14T10:49:38.145+0100 I CONTROL  [main] ** WARNING: Access control is not enabled for the database.
2023-03-14T10:49:38.145+0100 I CONTROL  [main] **          Read and write access to data and configuration is unrestricted.
2023-03-14T10:49:38.147+0100 I CONTROL  [main]
2023-03-14T10:49:38.148+0100 I CONTROL  [main] ** WARNING: This server is bound to localhost.
2023-03-14T10:49:38.148+0100 I CONTROL  [main] **          Remote systems will be unable to connect to this server.
2023-03-14T10:49:38.149+0100 I CONTROL  [main] **          Start the server with --bind_ip <address> to specify which IP
2023-03-14T10:49:38.150+0100 I CONTROL  [main] **          addresses it should serve responses from, or with --bind_ip_all to
2023-03-14T10:49:38.150+0100 I CONTROL  [main] **          bind to all interfaces. If this behavior is desired, start the
2023-03-14T10:49:38.151+0100 I CONTROL  [main] **          server with --bind_ip 127.0.0.1 to disable this warning.
2023-03-14T10:49:38.151+0100 I CONTROL  [main]
2023-03-14T10:49:38.154+0100 I SHARDING [mongosMain] mongos version v4.0.28
2023-03-14T10:49:38.154+0100 I CONTROL  [mongosMain] git version: af1a9dc12adcfa83cc19571cb3faba26eeddac92
2023-03-14T10:49:38.154+0100 I CONTROL  [mongosMain] allocator: tcmalloc
2023-03-14T10:49:38.155+0100 I CONTROL  [mongosMain] modules: none
2023-03-14T10:49:38.157+0100 I CONTROL  [mongosMain] build environment:
2023-03-14T10:49:38.158+0100 I CONTROL  [mongosMain]     distmod: 2008plus-ssl
2023-03-14T10:49:38.159+0100 I CONTROL  [mongosMain]     distarch: x86_64
2023-03-14T10:49:38.160+0100 I CONTROL  [mongosMain]     target_arch: x86_64
2023-03-14T10:49:38.161+0100 I CONTROL  [mongosMain] db version v4.0.28
2023-03-14T10:49:38.161+0100 I CONTROL  [mongosMain] git version: af1a9dc12adcfa83cc19571cb3faba26eeddac92
2023-03-14T10:49:38.162+0100 I CONTROL  [mongosMain] allocator: tcmalloc
```

Connectez-vous au serveur router (ConfigServers) avec le port de 27018 la commande ci-dessous.

```
C:\WINDOWS\system32>mongo --port 27018
MongoDB shell version v4.0.28
connecting to: mongodb://127.0.0.1:27018/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("d6a154b8-3622-4900-b9d0-36a021409e82") }
MongoDB server version: 4.0.28
Server has startup warnings:
2023-03-14T10:49:38.145+0100 I CONTROL  [main]
2023-03-14T10:49:38.145+0100 I CONTROL  [main] ** WARNING: Access control is not enabled for the database.
2023-03-14T10:49:38.145+0100 I CONTROL  [main] **          Read and write access to data and configuration is unrestric
ed.
2023-03-14T10:49:38.147+0100 I CONTROL  [main]
2023-03-14T10:49:38.148+0100 I CONTROL  [main] ** WARNING: This server is bound to localhost.
2023-03-14T10:49:38.148+0100 I CONTROL  [main] **          Remote systems will be unable to connect to this server.
2023-03-14T10:49:38.149+0100 I CONTROL  [main] **          Start the server with --bind_ip <address> to specify which I

2023-03-14T10:49:38.150+0100 I CONTROL  [main] **          addresses it should serve responses from, or with --bind_ip_
ll to
2023-03-14T10:49:38.150+0100 I CONTROL  [main] **          bind to all interfaces. If this behavior is desired, start t
e
2023-03-14T10:49:38.151+0100 I CONTROL  [main] **          server with --bind_ip 127.0.0.1 to disable this warning.
2023-03-14T10:49:38.151+0100 I CONTROL  [main]
mongos> sh.addShard("sh1/localhost:27030")
```

Maintenant le router de requête mongos est capable de communiquer avec les serveurs de configuration.

Remarque : Il est possible de rajouter chacun des serveurs du ReplicaSet, en les séparant par des virgules en faisant comme suit :

```
configReplSet/localhost:27017,localhost:27020,localhost:27021
```

Etape 8 : Ajouter des fragments à mongos
Nous devons activer le partitionnement (Sharding) afin que mongos sachent quels serveurs hébergeront les données distribuées et où se trouve un document.
Les shards peuvent alors être ajoutes les uns après les autres au niveau du mongos en mode console

On se connecte au mongos puis on ajoute chaque fragment individuellement

```
sh.addShard( "sh1/localhost:27030");
sh.addShard( "sh2/localhost:27031");
sh.addShard( "sh3/localhost:27032");
```

Démonstration :

Pour le sh.addShard("sh1/localhost:27030");

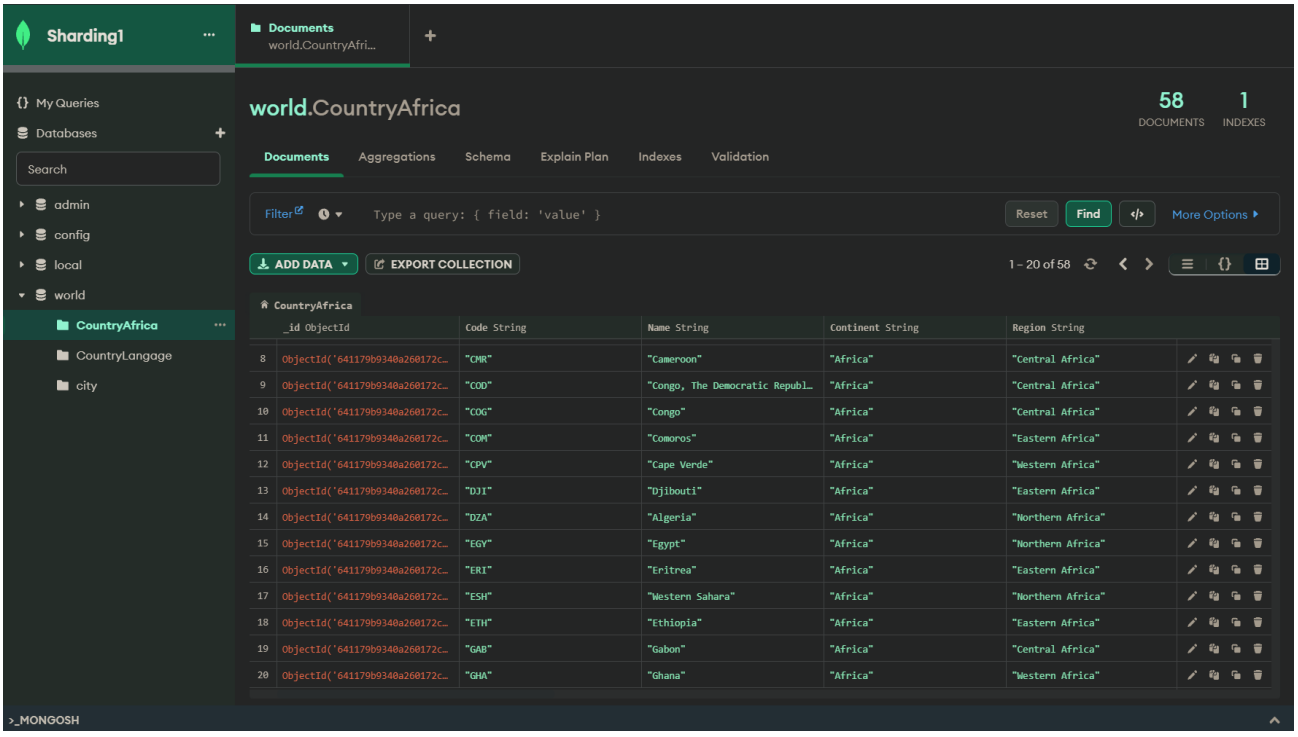
```
mongos> sh.addShard("sh1/localhost:27030")
{
        "shardAdded" : "sh1",
        "ok" : 1,
        "operationTime" : Timestamp(1678790387, 6),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1678790387, 6),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
```

Pour le sh.addShard("sh2/localhost:27031");

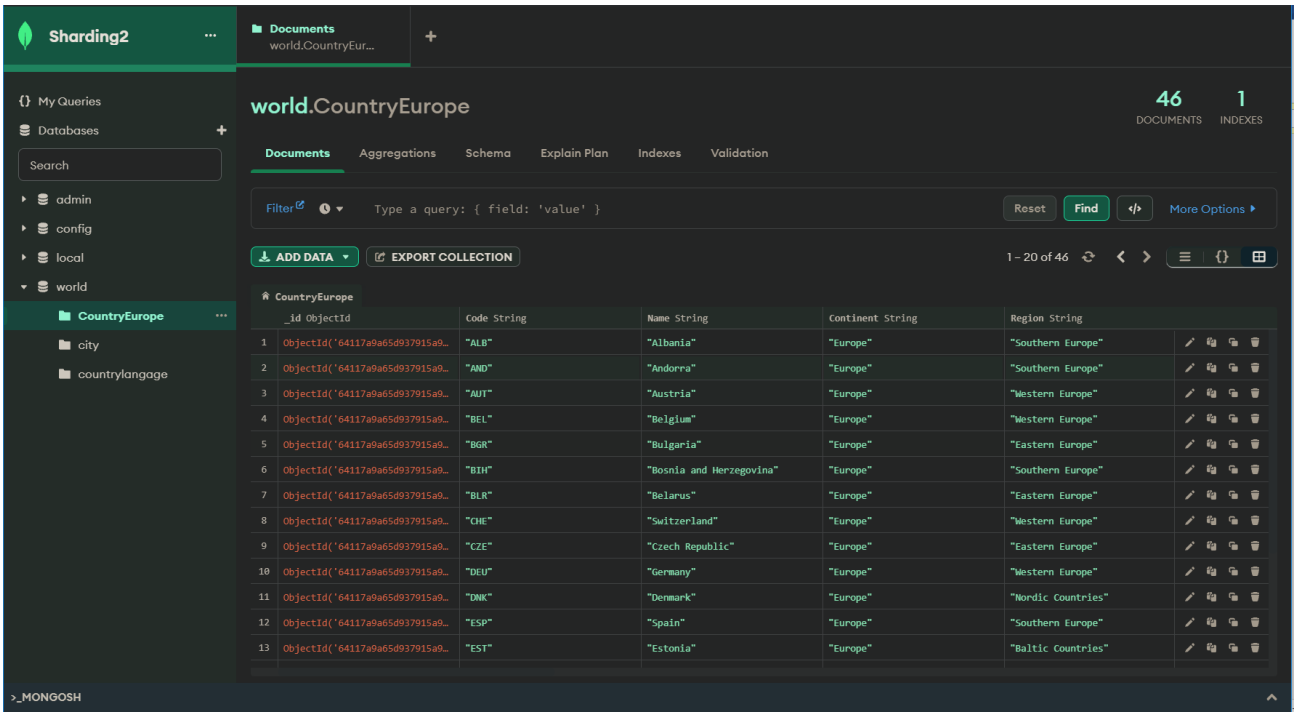
Pour le sh.addShard("sh3/localhost:27032");

Ça y est, l'architecture de distribution est mise en place. Il suffit maintenant de définir la collection que l'on veut distribuer. Pour cela, nous allons créer une base « world » et une collection « country ».
Maintenant nous avons configurer les shardings dans le routeur nous pouvons faire la distribution de la base de données en mettant sur les trois shardings les bases de données respectives :

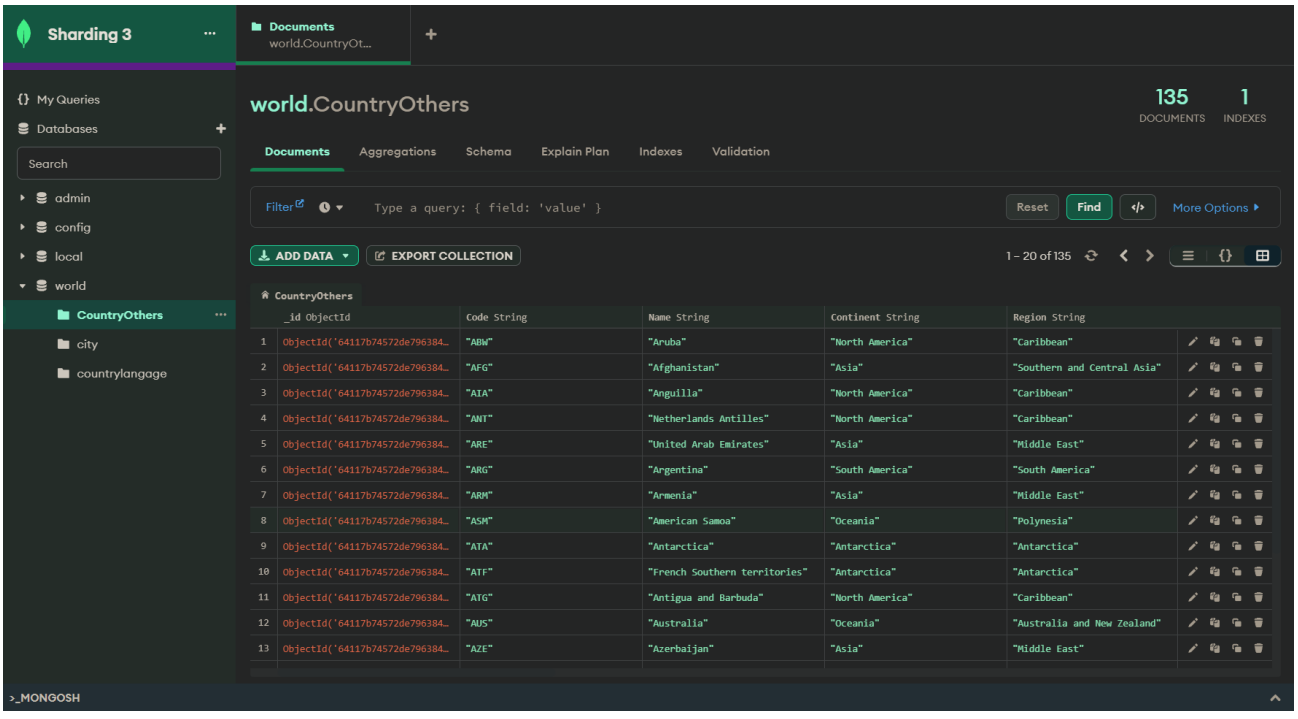
Sharding1 : Country (Afrique) qui écoute le port 27030



Sharding2 : Country (Europe) qui écoute le port 27031



Sharding3 : Country (Autres) qui écoute le port 27032



Pour de meilleures performances, sur chaque nœud il doit être prévu sur chacun de répartir la liste des pays sur des partitions par du sharding.
Après démonstration nous sommes finalement arrivées à conclure que la première partition du **sharding1** contient les pays d'Afrique, la seconde partition du **sharding2** contient les pays d'Europe, et la 3e partition du **sharding3** contient tous les autres pays.
Nous avons vu comment créer une architecture de distribution et comment distribuer une collection.