

1) Créer les répertoires de données

Dans le terminal CLIENT :

Trois répertoires de stockage ont été créés afin d'isoler les données du serveur de configuration et celles des deux shards.

```
C:\Users\Kaoutar>mkdir configsvrdb serv1 serv2  
C:\Users\Kaoutar>
```

2. Mise en place du Config Server

Terminal : CONFIG_SVR



```
Microsoft Windows [version 10.0.22631.5039]  
(c) Microsoft Corporation. Tous droits réservés.  
C:\Users\Kaoutar>cd /d C:\mongo-lab  
C:\mongo-lab>mongod --configsvr --replSet replicaconfig --dbpath .\configsvrdb --port 27019  
{"t": {"$date": "2025-12-17T22:23:12.762+01:00"}, "s": "I", "c": "-", "id": 8991200, "ctx": "thread1", "msg": "Shuffling initializers", "attr": {"seed": 2631359312}}  
{"t": {"$date": "2025-12-17T22:23:12.780+01:00"}, "s": "I", "c": "CONTROL", "id": 97374, "ctx": "thread1", "msg": "Automatically disabling TLS 1.0 and TLS 1.1, to force-enable TLS 1.1 specify --sslDisabledProtocols 'TLS1_0'; to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"},  
{"t": {"$date": "2025-12-17T22:23:12.925+01:00"}, "s": "I", "c": "NETWORK", "id": 4915701, "svc": "-", "ctx": "thread1", "msg": "Initialized wire specification", "attr": {"spec": {"incomingExternalClient": {"minWireVersion": 0, "maxWireVersion": 27}, "incomingInternalClient": {"minWireVersion": 0, "maxWireVersion": 27}, "outgoing": {"minWireVersion": 6, "maxWireVersion": 27}, "isInternalClient": true}}},  
{"t": {"$date": "2025-12-17T22:23:12.926+01:00"}, "s": "I", "c": "REPL", "id": 5123008, "svc": "-", "ctx": "thread1", "msg": "Successfully registered PrimaryOnlyService", "attr": {"service": "ReshardingCoordinatorService", "namespace": "config.reshardingOperations"}},  
{"t": {"$date": "2025-12-17T22:23:12.926+01:00"}, "s": "I", "c": "REPL", "id": 5123008, "svc": "-", "ctx": "thread1", "msg": "
```

Dans le terminal CLIENT :

Un serveur de configuration a été démarré sur le port 27019. Celui-ci est intégré dans un replica set nommé replicaconfig, indispensable au bon fonctionnement du sharding, car il stocke les métadonnées de partitionnement du cluster.

```

mongosh mongodb://127.0.0.1:27019
Using MongoDB: 8.2.2
Using Mongosh: 2.5.10
For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/
-----
The server generated these startup warnings when booting
2025-12-17T22:47:09.949+01:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2025-12-17T22:47:09.950+01:00: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning
-----
test> rs.initiate()
{
  info2: 'no configuration specified. Using a default configuration for the set',
  me: 'localhost:27019',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1766008085, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAA='),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1766008085, i: 1 })
}
replicaconfig [direct: secondary] test>

```

3. Démarrage du routeur mongos

Terminal : MONGOS

Le routeur mongos a été lancé. Il joue le rôle d'intermédiaire entre les clients et les shards, en redirigeant les requêtes vers les serveurs appropriés en fonction de la clé de sharding.

```

MONGOS
Microsoft Windows [version 10.0.22631.5039]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Kaoutar>mongos --configdb replicaconfig/localhost:27019
{"t": {"$date": "2025-12-17T22:29:04.866+01:00"}, "s": "I", "c": "-", "id": 8991200, "ctx": "thread1", "msg": "Shuffling initializers", "attr": {"seed": 4280342242}}
{"t": {"$date": "2025-12-17T22:29:04.874+01:00"}, "s": "W", "c": "SHARDING", "id": 24132, "ctx": "thread1", "msg": "Running a sharded cluster with fewer than 3 config servers should only be done for testing purposes and is not recommended for production."}
{"t": {"$date": "2025-12-17T22:29:04.988+01:00"}, "s": "I", "c": "CONTROL", "id": 97374, "ctx": "thread1", "msg": "Automatically disabling TLS 1.0 and TLS 1.1, to force-enable TLS 1.1 specify --sslDisabledProtocols 'TLS1_0'; to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t": {"$date": "2025-12-17T22:29:05.006+01:00"}, "s": "I", "c": "NETWORK", "id": 4915701, "ctx": "thread1", "msg": "Initialized wire specification", "attr": {"spec": {"incomingExternalClient": {"minWireVersion": 0, "maxWireVersion": 27}, "incomingInternalClient": {"minWireVersion": 0, "maxWireVersion": 27}, "outgoing": {"minWireVersion": 27, "maxWireVersion": 27}, "isInternalClient": true}}}
{"t": {"$date": "2025-12-17T22:29:05.014+01:00"}, "s": "I", "c": "HEALTH", "id": 5936503, "ctx": "thread1", "msg": "Fault manager changed state", "attr": {"state": "StartupCheck"}}
{"t": {"$date": "2025-12-17T22:29:05.017+01:00"}, "s": "W", "c": "CONTROL", "id": 22120, "ctx": "thread1", "msg": "Access control is not enabled for the database. Read and write access to data and configuration is unrestricted", "tags": "tags"

```

Dans client:

```

mongosh mongodb://127.0.0.1:27017/test
Using MongoDB: 8.2.2
Using Mongosh: 2.5.10

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

-----
  The server generated these startup warnings when booting
  2025-12-17T22:50:02.869+01:00: Access control is not enabled for the database. Read and write a
  configuration is unrestricted
  2025-12-17T22:50:02.870+01:00: This server is bound to localhost. Remote systems will be unable
  to connect. Start the server with --bind_ip <address> to specify which IP addresses it should serve res
  bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_
  this warning
-----

test> rs.initiate()
{
  info2: 'no configuration specified. Using a default configuration for the set',
  me: 'localhost:20004',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1766008395, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAA='),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1766008395, i: 1 })
}
replicashard1 [direct: secondary] test>

```

4. Mise en place des shards

◆ Shard 1

Terminal : SHARD1

```

SHARD1
Microsoft Windows [version 10.0.22631.5039]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Kaoutar>mongod --replSet replicashard1 --dbpath serv1 --shardsvr --port 20004
{"t": {"$date": "2025-12-17T22:30:50.786+01:00"}, "s": "I", "c": "-", "id": 8991200, "ctx": "thread1", "msg": "Shuffling
initializers", "attr": {"seed": 1804748232}}
{"t": {"$date": "2025-12-17T22:30:50.804+01:00"}, "s": "I", "c": "CONTROL", "id": 97374, "ctx": "thread1", "msg": "Automatically
disabling TLS 1.0 and TLS 1.1, to force-enable TLS 1.1 specify --sslDisabledProtocols 'TLS1_0'; to force-enable TLS
1.0 specify --sslDisabledProtocols 'none'"}
{"t": {"$date": "2025-12-17T22:30:50.937+01:00"}, "s": "I", "c": "NETWORK", "id": 4915701, "svc": "-", "ctx": "thread1", "msg": "
Initialized wire specification", "attr": {"spec": {"incomingExternalClient": {"minWireVersion": 0, "maxWireVersion": 27}, "inco
mingInternalClient": {"minWireVersion": 0, "maxWireVersion": 27}, "outgoing": {"minWireVersion": 6, "maxWireVersion": 27}, "isInte
rnalClient": true}}}
{"t": {"$date": "2025-12-17T22:30:50.941+01:00"}, "s": "I", "c": "REPL", "id": 5123008, "svc": "-", "ctx": "thread1", "msg": "
Successfully registered PrimaryOnlyService", "attr": {"service": "ShardingDDLCoordinator", "namespace": "config.system.shard
ing_ddl_coordinators"}}
{"t": {"$date": "2025-12-17T22:30:50.941+01:00"}, "s": "I", "c": "REPL", "id": 5123008, "svc": "-", "ctx": "thread1", "msg": "
Successfully registered PrimaryOnlyService", "attr": {"service": "RenameCollectionParticipantService", "namespace": "config.
localRenameParticipants"}}
{"t": {"$date": "2025-12-17T22:30:50.941+01:00"}, "s": "I", "c": "REPL", "id": 5123008, "svc": "-", "ctx": "thread1", "msg": "
Successfully registered PrimaryOnlyService", "attr": {"service": "ReshardingDonorService", "namespace": "config.localReshard
Operations_donor"}}

```

Dans CLIENT :

```
Client x + v
Microsoft Windows [version 10.0.22631.5039]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Kaoutar>mongosh --port 27019
Current Mongosh Log ID: 694320352e24a238d01e2620
Connecting to:      mongodb://127.0.0.1:27019/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.10
Using MongoDB:     8.2.2
Using Mongosh:    2.5.10

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
  2025-12-17T22:23:13.194+01:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
  2025-12-17T22:23:13.195+01:00: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning
-----
test> mongo --port 20004
... rs.initiate()
...
```

◆ Shard 2

Terminal : SHARD2

```
SHARD2 x + v
Microsoft Windows [version 10.0.22631.5039]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Kaoutar>mongod --replSet replicashard2 --dbpath serv2 --shardsvr --port 20005
{"t": {"$date": "2025-12-17T22:32:35.405+01:00"}, "s": "I", "c": "-", "id": 8991200, "ctx": "thread1", "msg": "Shuffling initializers", "attr": {"seed": 1760411379}}
{"t": {"$date": "2025-12-17T22:32:35.514+01:00"}, "s": "I", "c": "CONTROL", "id": 97374, "ctx": "thread1", "msg": "Automatically disabling TLS 1.0 and TLS 1.1, to force-enable TLS 1.1 specify --sslDisabledProtocols 'TLS1_0'; to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t": {"$date": "2025-12-17T22:32:35.521+01:00"}, "s": "I", "c": "NETWORK", "id": 4915701, "svc": "-", "ctx": "thread1", "msg": "Initialized wire specification", "attr": {"spec": {"incomingExternalClient": {"minWireVersion": 0, "maxWireVersion": 27}, "incomingInternalClient": {"minWireVersion": 0, "maxWireVersion": 27}, "outgoing": {"minWireVersion": 6, "maxWireVersion": 27}, "isInternalClient": true}}}
{"t": {"$date": "2025-12-17T22:32:35.523+01:00"}, "s": "I", "c": "REPL", "id": 5123008, "svc": "-", "ctx": "thread1", "msg": "Successfully registered PrimaryOnlyService", "attr": {"service": "ShardingDDLCoordinator", "namespace": "config.system.sharding_ddl_coordinators"}}
{"t": {"$date": "2025-12-17T22:32:35.523+01:00"}, "s": "I", "c": "REPL", "id": 5123008, "svc": "-", "ctx": "thread1", "msg": "Successfully registered PrimaryOnlyService", "attr": {"service": "RenameCollectionParticipantService", "namespace": "config.localRenameParticipants"}}
```

Dans CLIENT :

```
mongosh mongodb://127.0.0.1:27017/test + v
Using MongoDB: 8.2.2
Using Mongosh: 2.5.10

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

-----
The server generated these startup warnings when booting
2025-12-17T22:56:23.198+01:00: Access control is not enabled for the database.
iguration is unrestricted
2025-12-17T22:56:23.199+01:00: This server is bound to localhost. Remote system
rver. Start the server with --bind_ip <address> to specify which IP addresses it s
bind_ip_all to bind to all interfaces. If this behavior is desired, start the serv
this warning
-----

test> rs.initiate()
{
  info2: 'no configuration specified. Using a default configuration for the set',
  me: 'localhost:20005',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1766008636, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAA='),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1766008636, i: 1 })
}
replicashard2 [direct: secondary] test>
```

Deux shards ont été déployés, chacun configuré comme un replica set. Cette approche permet d'assurer la tolérance aux pannes et la montée en charge du cluster.

4. Ajout des shards au cluster

Terminal : CLIENT (connecté à mongos)

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&ssl=false
```

Microsoft Windows [version 10.0.22631.5039]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Kaoutar>mongosh --port 27017
Current Mongosh Log ID: 694327baa25518a6a71e2620
Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&ssl=false
Using MongoDB: 8.2.2
Using Mongosh: 2.5.10

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

The server generated these startup warnings when booting
2025-12-17T22:07:30.903+01:00: Access control is not enabled for the configuration is unrestricted

```
test> db.isMaster()  
{  
  ismaster: true,  
  topologyVersion: {  
    processId: ObjectId('69431b91e0dd838a2fdd659b'),  
    counter: Long('0')  
  },  
  maxBsonObjectSize: 16777216,  
  maxMessageSizeBytes: 48000000,  
  maxWriteBatchSize: 100000,  
  localTime: ISODate('2025-12-17T22:00:18.740Z'),  
  logicalSessionTimeoutMinutes: 30,
```

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&ssl=false
```

Using Mongosh: 2.5.10

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

The server generated these startup warnings when booting
2025-12-17T22:07:30.903+01:00: Access control is not enabled for the configuration is unrestricted

```
test> db.isMaster()  
{  
  ismaster: true,  
  topologyVersion: {  
    processId: ObjectId('69431b91e0dd838a2fdd659b'),  
    counter: Long('0')  
  },  
  maxBsonObjectSize: 16777216,  
  maxMessageSizeBytes: 48000000,  
  maxWriteBatchSize: 100000,  
  localTime: ISODate('2025-12-17T22:00:18.740Z'),  
  logicalSessionTimeoutMinutes: 30,  
  connectionId: 4,  
  minWireVersion: 0,  
  maxWireVersion: 27,  
  readOnly: false,  
  ok: 1,  
  isWritablePrimary: true  
}
```

test>

Les deux shards ont été ajoutés dynamiquement au cluster MongoDB à l'aide du routeur mongos.

```
{[direct: mongos] test> sh.addShard("replicashard1/localhost:20004")
{
  shardAdded: 'replicashard1',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1766009761, i: 57 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAA='),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1766009761, i: 57 })
}
[direct: mongos] test> |}

} {[direct: mongos] test> sh.addShard("replicashard2/localhost:20005")
{
  shardAdded: 'replicashard2',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1766009806, i: 14 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAA='),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1766009806, i: 14 })
}
[direct: mongos] test> |}
```

5. Activation du sharding sur la base

Le sharding n'étant pas activé par défaut, il a été explicitement activé sur la base de données mabasefilms.

```
{[direct: mongos] test> sh.enableSharding("mabasefilms")
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1766009855, i: 21 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAA='),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1766009855, i: 21 })
}
[direct: mongos] test> |}
```

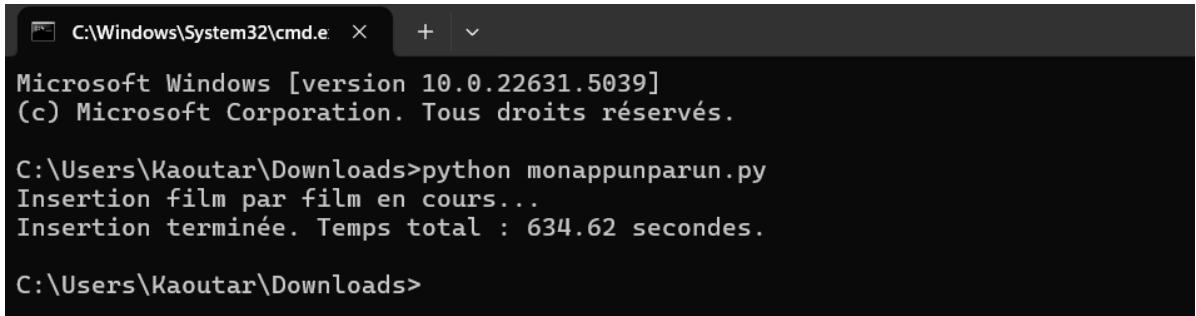
6. Sharding de la collection

La collection films a été shardée en utilisant le champ titre comme clé de partitionnement. Cette clé permet de répartir les documents entre les shards selon des plages de valeurs.

```
[direct: mongos] test> sh.shardCollection("mabasefilms.films", { titre: 1 })
{
  collectionsharded: 'mabasefilms.films',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1766009919, i: 38 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAA='),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1766009919, i: 37 })
}
[direct: mongos] test> |
```

7. Insertion des données et observation

Lors de l'insertion massive des films, MongoDB a automatiquement découpé les données en chunks et les a redistribués entre les shards afin d'équilibrer la charge, grâce au balancer.



```
C:\Windows\System32\cmd.exe + ^
Microsoft Windows [version 10.0.22631.5039]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Kaoutar\Downloads>python monappunparun.py
Insertion film par film en cours...
Insertion terminée. Temps total : 634.62 secondes.

C:\Users\Kaoutar\Downloads>
```

→ La commande sh.status() :

Cette commande permet de vérifier l'état général du cluster sharded, en affichant les shards, les chunks et l'activité du balancer.

```
replicaconfig [direct: primary] mabasefilms> sh.status()
Warning: MongoshWarning: [SHAPI-10003] You are not connected to a mongos. This command may not work as expected.
shardingVersion
{
  _id: ObjectId('69432515f31da09e1bb7763d'),
  clusterId: ObjectId('69432515f31da09e1bb7763c')
}
---
shards
[
  {
    _id: 'replicashard1',
    host: 'replicashard1/localhost:20004',
    state: 1,
    topologyTime: Timestamp({ t: 1766009761, i: 32 }),
    replSetConfigVersion: Long('-1')
  },
  {
    _id: 'replicashard2',
    host: 'replicashard2/localhost:20005',
    state: 1,
    topologyTime: Timestamp({ t: 1766009805, i: 32 }),
    replSetConfigVersion: Long('-1')
  }
]
---
active mongoses
[ { '8.2.2': 2 } ]
```

→ La commande db.printShardingStatus():

Cette commande fournit des informations détaillées sur le sharding de chaque collection, la clé de sharding utilisée, et la répartition des chunks entre les shards.

```
replicaconfig [direct: primary] mabasefilms> db.printShardingStatus()
shardingVersion
{
  _id: ObjectId('69432515f31da09e1bb7763d'),
  clusterId: ObjectId('69432515f31da09e1bb7763c')
}
---
shards
[
  {
    _id: 'replicashard1',
    host: 'replicashard1/localhost:20004',
    state: 1,
    topologyTime: Timestamp({ t: 1766009761, i: 32 }),
    replSetConfigVersion: Long('-1')
  },
  {
    _id: 'replicashard2',
    host: 'replicashard2/localhost:20005',
    state: 1,
    topologyTime: Timestamp({ t: 1766009805, i: 32 }),
    replSetConfigVersion: Long('-1')
  }
]
---
active mongoses
[ { '8.2.2': 2 } ]
```

→ La commande db.stats():

Cette commande affiche les statistiques globales de la base de données sélectionnée, notamment le nombre de documents, la taille des données et l'espace utilisé par les indexés.

```
replicaconfig [direct: primary] mabasefilms> db.stats()
{
  db: 'mabasefilms',
  collections: Long('0'),
  views: Long('0'),
  objects: Long('0'),
  avgObjSize: 0,
  dataSize: 0,
  storageSize: 0,
  indexes: Long('0'),
  indexSize: 0,
  totalSize: 0,
  scaleFactor: Long('1'),
  fsUsedSize: 0,
  fsTotalSize: 0,
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1766011310, i: 3 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAA='),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1766011310, i: 3 })
}
replicaconfig [direct: primary] mabasefilms>
```

1. Qu'est-ce que le sharding dans MongoDB et pourquoi est-il utilisé ?

Le sharding est le partitionnement horizontal des données pour répartir la charge et gérer de grands volumes de données de manière scalable.

2. Quelle est la différence entre le sharding et la réPLICATION dans MongoDB ?

La réPLICATION assure la redondance et la haute disponibilité des données, tandis que le sharding distribue les données sur plusieurs serveurs pour l'évolutivité.

3. Quels sont les composants d'une architecture shardée (mongos, config servers, shards) ?

Les shards stockent les données, les config servers conservent la configuration du cluster, et les mongos route les requêtes vers les shards appropriés.

4. Quelles sont les responsabilités des config servers (CSRS) dans un cluster shardé ?

Ils stockent la métadonnée du cluster, la carte des chunks et la configuration du sharding.

5. Quel est le rôle du mongos router ?

Le mongos dirige les requêtes clients vers les shards corrects en utilisant les informations des config servers.

6. Comment MongoDB décide-t-il sur quel shard stocker un document ?

MongoDB utilise la clé de sharding pour déterminer le chunk correspondant et stocke le document sur le shard qui possède ce chunk.

7. Qu'est-ce qu'une clé de sharding et pourquoi est-elle essentielle ?

C'est un ou plusieurs champs définissant la distribution des documents entre shards, essentielle pour équilibrer la charge et optimiser les requêtes.

8. Quels sont les critères de choix d'une bonne clé de sharding ?

Cardinalité élevée, uniformité de distribution, stabilité et pertinence par rapport aux requêtes fréquentes.

9. Qu'est-ce qu'un chunk dans MongoDB ?

Un chunk est un sous-ensemble contigu de documents dans une collection, basé sur la clé de sharding.

10. Comment fonctionne le splitting des chunks ?

MongoDB divise automatiquement un chunk lorsqu'il dépasse une taille limite (par défaut 64 Mo) pour équilibrer les données.

11. Que fait le balancer dans un cluster shardé ?

Il redistribue les chunks entre shards pour équilibrer la charge et optimiser l'espace de stockage.

12. Quand et comment le balancer déplace-t-il des chunks ?

Lorsqu'un shard devient surchargé, le balancer migre des chunks vers des shards moins chargés, automatiquement en arrière-plan.

13. Qu'est-ce qu'un hot shard et comment l'éviter ?

Un shard qui reçoit trop de trafic ou de données ; on l'évite en choisissant une clé de sharding bien distribuée.

14. Quels problèmes une clé de sharding monotone peut-elle engendrer ?

Elle peut créer un hotspot, concentrer l'écriture sur un seul shard et réduire les performances.

15. Comment activer le sharding sur une base de données et sur une collection ?

```
sh.enableSharding("mabasefilms")
sh.shardCollection("mabasefilms.films", { "titre": 1 })
```

16. Comment ajouter un nouveau shard à un cluster MongoDB ?

```
sh.addShard("replicaName/host:port")
```

17. Comment vérifier l'état du cluster shardé (commandes usuelles) ?

```
sh.status(), db.printShardingStatus(), db.stats()
```

18. Dans quels cas faut-il envisager d'utiliser un hashed sharding key ?

Lorsque la distribution uniforme des documents est prioritaire, indépendamment de la valeur logique des données.

19. Dans quels cas faut-il privilégier un ranged sharding key ?

Lorsque les requêtes sont souvent basées sur des plages de valeurs et que l'ordre des données est important.

20. Qu'est-ce que le zone sharding et quel est son intérêt ?

C'est l'assignation de chunks à des zones spécifiques pour contrôler la localisation des données et optimiser la géodistribution ou la conformité.

21. Comment MongoDB gère-t-il les requêtes multi-shards ?

Le mongos divise la requête et interroge tous les shards concernés, puis combine les résultats avant de les renvoyer au client.

22. Comment optimiser les performances de requêtes dans un environnement shardé ?

Utiliser des clés de sharding adaptées, créer des indexées sur la clé de sharding et limiter les requêtes aux shards ciblés.

23. Que se passe-t-il lorsqu'un shard devient indisponible ?

Les données sur ce shard deviennent inaccessibles, mais les autres shards restent opérationnels ; la réplication peut assurer la continuité si configurée.

24. Comment migrer une collection existante vers un schéma shardé ?

Créer la collection shardée avec sh.shardCollection() et laisser MongoDB redistribuer les documents selon la clé de sharding.

25. Quels outils ou métriques utiliser pour diagnostiquer les problèmes de sharding ?

sh.status(), db.printShardingStatus(), db.currentOp(), logs MongoDB, et monitoring via MongoDB.