

# Migration MMapV1 vers Wired Tiger

## Sommaire

- Concept
- Préparation
  - Quel format de données MongoDB utilise?
  - Limiter les données sauvegardées en base
  - Faire un état des lieux
- Migration de la base
  - Arrêt de Shinken
  - Sauvegarde des données
  - Extraire les données à migrer
  - Arrêt de mongod et purge des données
    - Debian 13
    - RHEL / CentOS 7 ou RHEL / Alma / Rocky 8 ou RHEL / Alma / Rocky 9
  - Changement du moteur de MongoDB vers wiredTiger
  - Migration des données
  - Vérification
  - Démarrer Shinken

## Concept

À l'origine, MongoDB utilisait le moteur MMAPv1 pour gérer le stockage de ses données. Ce dernier ne gérait pas la compression des données.

Le moteur wiredTiger, plus récent, intègre nativement des fonctionnalités de compression.

Cela permet de réduire significativement l'empreinte disque de MongoDB, et d'optimiser l'utilisation de l'espace de stockage.

## Préparation

### Quel format de données MongoDB utilise?

Vérifier que le type de moteur de stockage est bien MMapv1 à l'aide de la commande suivante :

```
mongo shinken --quiet --eval "print(db.serverStatus().storageEngine.name)"
```

Le retour de la commande doit être **MMapv1**.



Si le retour est **wiredTiger**, alors la base de données est déjà migrée et il n'est pas nécessaire de continuer cette procédure.

### Limiter les données sauvegardées en base

Plus la base de données contient de données, plus la migration vers un nouveau moteur de données prendra du temps.

Pour cette raison, il peut être judicieux de limiter les données stockées en base avant de procéder à la migration ( voir la page [MongoDB - Méthode 1 : Ne garder que les données pertinentes](#) ).

### Faire un état des lieux

En premier lieu, il est important de récupérer des statistiques sur la base de données qui serviront, plus tard, à vérifier que la migration de données s'est bien passée.

```
mongo shinken --quiet --eval 'db.getCollectionNames()' | tee db_info.info
mongo shinken --quiet --eval 'printjson(db.adminCommand( { listDatabases: 1 } ));' | tee -a db_info.info
```

Ces commandes vont afficher et conserver dans le fichier db\_info.info respectivement :

- La liste des collections de la base *shinken* dans une chaîne de caractères séparée par des virgules.
- La liste des bases présentes sur le serveur, dans un format JSON.

En fonction de l'architecture, il est possible que ces commandes ne retournent rien.

Par exemple, exécuter ces commandes sur un serveur qui héberge uniquement la base du Synchronizer ne donnera un résultat que pour la deuxième commande.



Si les deux commandes ne retournent rien, s'assurer qu'elles sont exécutées sur le bon serveur.

Cette procédure changera bien le type du moteur de stockage, mais aucune donnée ne sera impactée.

## Migration de la base

### Arrêt de Shinken

Il faut éteindre Shinken pour continuer la procédure

```
service-shinken stop
```

### Sauvegarde des données



#### Important !

Avant toute opération, faire un shinken-backup complet ( *ou avec les options qui permettent de sauvegarder les données* ) du serveur impacté.

### Extraire les données à migrer

On enregistre les données à migrer :

```
cd /tmp  
mongodump -o data_to_migrate
```

### Arrêt de mongod et purge des données

#### Debian 13

```
service mongod stop  
cd /var/lib/mongodb  
rm -fr *
```

#### RHEL / CentOS 7 ou RHEL / Alma / Rocky 8 ou RHEL / Alma / Rocky 9

```
service mongod stop  
cd /var/lib/mongo  
rm -fr *
```

### Changement du moteur de MongoDB vers wiredTiger

Changer le moteur de **mmapv1** à **wiredTiger**.

Mettre à jour le fichier `/etc/mongod.conf` :

#### `/etc/mongod.conf`

```
storage:  
  engine: wiredTiger
```

Démarrer Mongo

```
service mongod start
```

Vérifier le type de moteur de stockage :

```
mongo shinken --quiet --eval "print(db.serverStatus().storageEngine.name)"
```

Le retour de la commande doit être **wiredTiger**

## Migration des données

On restaure les données dans MongoDB

```
cd /tmp  
mongorestore data_to_migrate
```

## Vérification

On vérifie que les données obtenues correspondent à celles du fichier db\_info.info que l'on a créé précédemment.

```
mongo shinken --quiet --eval 'db.getCollectionNames()'  
mongo shinken --quiet --eval 'printjson(db.adminCommand( { listDatabases: 1 } ))';'
```

## Démarrer Shinken

```
service-shinken start
```