

MongoDB - surveillance du taux de fragmentation de la base

Sommaire

[Surveillance du taux de fragmentation de la base](#)

[Paramètres d'exécution](#)

[Options génériques](#)

[Options de connexion SSH](#)

[Options d'authentification](#)

[Options SSL/TLS](#)

[Données retournées](#)

[Exécution sur une base avec MMapV1](#)

[Exécution sur une base avec Wired Tiger](#)

Surveillance du taux de fragmentation de la base

Au fur et à mesure des insertions / suppressions d'éléments en base de données, l'espace de stockage peut se retrouver morcelé.

L'enchevêtrement des zones d'espace libre et des zones contenant des données est tel, qu'il devient difficile de réaffecter les zones libres pour y stocker de nouvelles données.

Le ratio espace de stockage utilisé par rapport à la taille effective des données devient plus important, c'est ce qu'on appelle la fragmentation.

Il est possible de surveiller ce taux de fragmentation et même le réduire avec la commande suivante :

- [check_mongo_fragmentation.py](#)

Ce script permet d'avoir le calcul de ce taux, avec les volumes utilisés et les volumes sur disque.



Attention : ce script ne fonctionne que sur installation de Shinken Entreprise supérieure ou égale à la V02.08.02-RC019

Paramètres d'exécution

Sans paramètre, la commande se connecte au serveur MongoDB local.

La commande dispose d'options de connexion à la base MongoDB qui peuvent être utilisés dans les cas suivants :

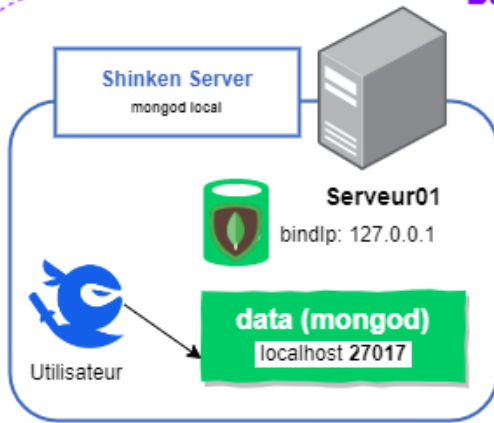
- La base de données MongoDB ne se trouve pas sur la machine qui exécute la commande.
- L'authentification par mot de passe à la base MongoDB est activée.
- Le port de MongoDB n'est pas celui par défaut (*défaut : 27017*).



La combinaison des options de connexion à MongoDB peut rapidement devenir complexe ; voici des paramètres adaptés aux cas les plus courants.

Options génériques

Base MongoDB en local

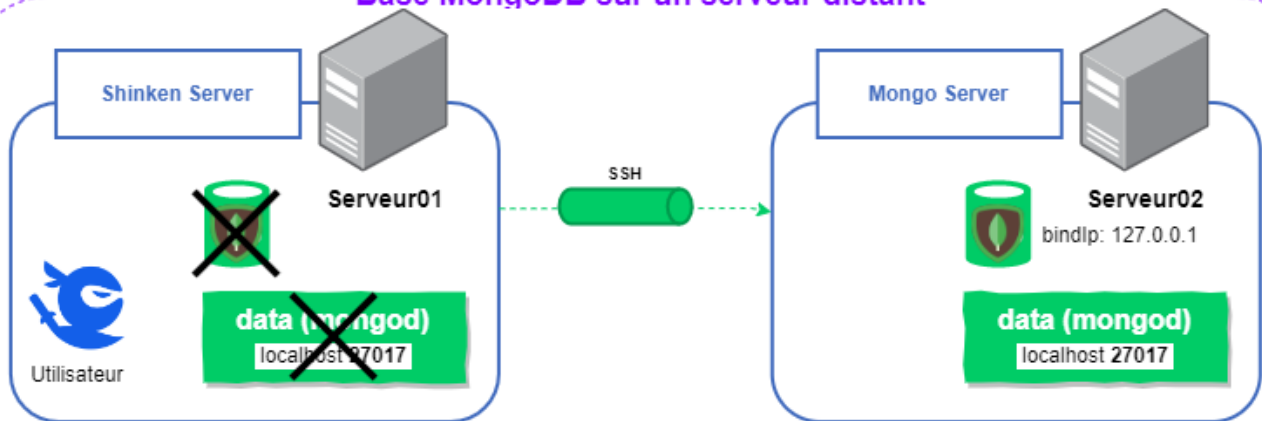


```
[root@serveur01 ~] shinken-commande --mongo-host 127.0.0.1 --mongo-port 27017 --mongo-database shinken
```

| Option | Valeur par défaut | Description |
|------------------------------------|--|--|
| --mongo-host <i>ARG</i> | localhost | Nom ou IP du serveur MongoDB. |
| --mongo-port <i>A</i> <i>RG</i> | 27017 | Port de la base MongoDB. |
| --mongo-database <i>ARG</i> | shinken (ou synchronizer si la commande concerne la base du Synchronizer) | Nom de la base de données à utiliser dans MongoDB. À n'utiliser que si la configuration du module ou du démon a changé la base utilisée par défaut. |

Options de connexion SSH

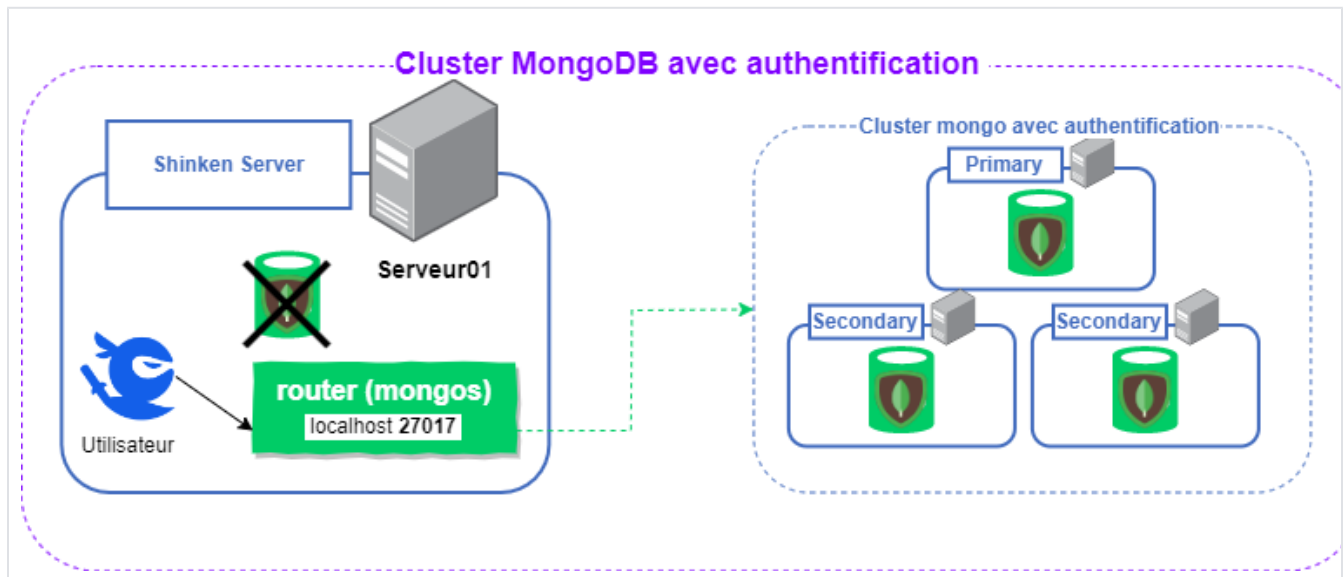
Base MongoDB sur un serveur distant



```
[root@serveur01 ~] shinken-command --mongo-host serveur02 --mongo-port 27017 --mongo-use-ssh --mongo-ssh-key /var/lib/shinken/.ssh/id_rsa --mongo-ssh-user shinken
```

| Option | Valeur par défaut | Description |
|-----------------------------------|---|---|
| <code>--mongo-use-ssh</code> | --- | Active la connexion SSH au serveur MongoDB. |
| <code>--mongo-ssh-key ARG</code> | <code>/var/lib/shinken/.ssh/id_rsa</code> | Clé privée SSH pour la connexion au serveur MongoDB. À utiliser en complément de l'option <code>--mongo-use-ssh</code> . |
| <code>--mongo-ssh-user ARG</code> | <code>shinken</code> | Utilisateur à utiliser pour la connexion SSH. À utiliser en complément de l'option <code>--mongo-use-ssh</code> . |

Options d'authentification

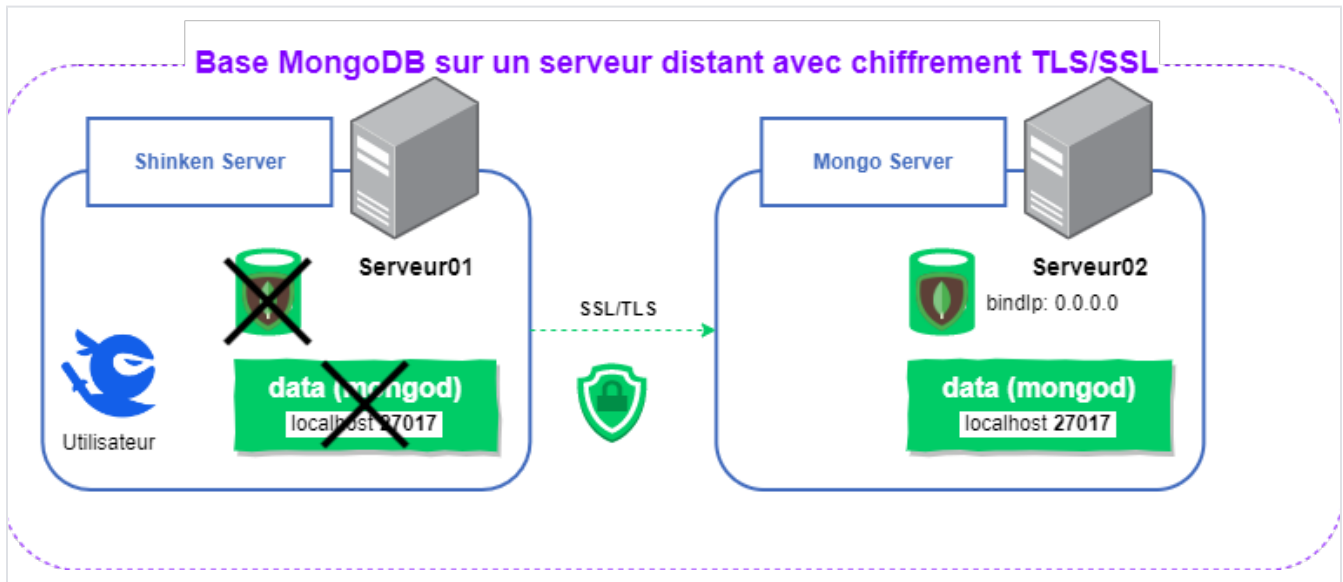


```
[root@serveur01 ~] shinken-command --mongo-host 127.0.0.1 --mongo-port 27017 --mongo-username shinken --mongo-password shinken
```

| Option | Valeur par défaut | Description |
|-----------------------------------|-------------------|--|
| <code>--mongo-username ARG</code> | --- | Utilisateur pour l'authentification avec mot de passe. |

| | | |
|------------------------------------|----------------|---|
| <pre>-- mongo - password ARG</pre> | <pre>---</pre> | <p>Mot de passe de l'utilisateur pour l'authentification avec mot de passe.</p> <p>À utiliser en complément de l'option <code>--mongo-username</code>.</p> <div style="border: 1px solid green; padding: 10px; margin: 10px 0;"> <p>✔ Si l'option <code>--mongo-password</code> est utilisée, le mot de passe risque d'être visible dans l'historique des commandes (<i>via la commande <code>history</code></i>).</p> <p>Pour éviter d'exposer le mot de passe, il est possible d'utiliser cette commande uniquement avec l'option <code>--mongo-username</code>. Un prompt interactif apparaîtra alors pour demander le mot de passe.</p> <p>Pour automatiser les commandes dans un script, il est possible de rediriger le contenu d'un fichier contenant le mot de passe (<i>par exemple : <code>--mongo-password \$(cat my_file_with_password)</code></i>).</p> </div> |
|------------------------------------|----------------|---|

Options SSL/TLS



```
[root@serveur01 ~] shinken-command --mongo-host serveur02 --mongo-port 27017 --mongo-ssl-ca-file /etc/shinken/certs/mongo/ca.pem --mongo-ssl-pem-key-file /etc/shinken/certs/mongo/client.pem
```

| Option | Valeur par défaut | Description |
|---|-------------------|--|
| <code>--mongo-ssl</code> | --- | Active SSL/TLS pour les communications avec la base MongoDB. |
| <code>--mongo-ssl-ca-file ARG</code> | --- | Chemin vers le fichier de l'autorité de certification (<i>CA</i>) utilisé pour vérifier le certificat SSL de MongoDB. À utiliser en complément de l'option <code>--mongo-ssl</code> . |
| <code>--mongo-ssl-pem-key-file ARG</code> | --- | Chemin vers le fichier contenant le certificat SSL du client. À utiliser en complément de l'option <code>--mongo-ssl</code> . |
| <code>--mongo-ssl-pem-key-password ARG</code> | --- | Mot de passe du certificat SSL du client. À utiliser en complément de l'option <code>--mongo-ssl</code> . |

| | | |
|---|-----|--|
| <code>--mongo-ssl-crl-file ARG</code> | --- | Chemin vers le fichier CRL (<i>liste de révocation</i>) des certificats SSL à rejeter. À utiliser en complément de l'option <code>--mongo-ssl</code> . |
| <code>--mongo-ssl-allow-invalid-hostnames</code> | --- | Accepter le certificat SSL de MongoDB même si le nom d'hôte du certificat ne correspond pas à celui du serveur. À utiliser en complément de l'option <code>--mongo-ssl</code> . |
| <code>--mongo-ssl-allow-invalid-certificates</code> | --- | Accepter le certificat SSL de MongoDB même s'il est invalide, par exemple expiré. À utiliser en complément de l'option <code>--mongo-ssl</code> . |

Données retournées

La commande va fournir les informations suivantes, pour chacune des bases de données présente sur le serveur :

- **Database:** le nom de la base
- **Disk-usage:** la consommation disque de la base
- **Data:** le volume de données contenu dans la base
- **Compression-save:** espace disque économisé grâce à la compression de données (*seulement pour Wired Tiger*)
- **Fragmented:** espace non utilisé dû à la fragmentation
 - Cet espace peut être réutilisé pour de nouvelles données, à la discrétion du moteur.
 - En cas de compactage ou migration, la majeure partie de cet espace pourra être récupéré.

Exécution sur une base avec MMapV1

Sur une base avec **MMapV1** l'exécution va donner un résultat suivant :

```
Mongodb storage engine: mmapv1
```

| Database | Disk-usage | Data | Compression-save | Fragmented (lost) |
|-------------------|------------|--------|------------------|-------------------|
| - OK synchronizer | 128.0Mo | 12.3Mo | 0.0o | 115.7Mo |
| - OK shinken | 32.0Mo | 3.2Ko | 0.0o | 32.0Mo |
| - OK local | 32.0Mo | 31.8Ko | 0.0o | 32.0Mo |
| - !! TOTAL | 192.0Mo | 12.4Mo | 0.0o | 179.6Mo (93%) |

```
--> You are still using the old mongo storage engine (MMapV1). You should migrate to WiredTiger instead to save space.
--> Fragmentation is very high: 179.6Mo is lost.
--> You can reclaim space by migrate to WiredTiger engine. Please look at the Shinken Enterprise documentation.
```

- Le script va conseiller de migrer de moteur de données, de **MMapV1** vers **Wired Tiger**, en demandant à se référer à la documentation (*voir la page MongoDB - maîtriser l'espace utilisé*).
- Le script conseillera une migration de la base de données uniquement si le pourcentage d'espace perdu est > **50%**
 - Dans ce cas, il conseille de se référer à la documentation (*voir la page MongoDB - maîtriser l'espace utilisé*).
- Il fournit l'espace perdu à cause de la fragmentation, qui pourra être récupéré lors d'une migration.
 - L'espace sera probablement encore plus grand lors de la migration, car **Wired Tiger** compresse les données sur disque.

Exécution sur une base avec Wired Tiger

Sur une base avec **Wired Tiger**, le résultat sera le suivant :

Mongodb storage engine: **wiredTiger**

| Database | Disk-usage | Data | Compression-save | Fragmented(lost) |
|----------------------------|------------|---------|------------------|------------------|
| - OK local | 36.0Ko | 5.2Ko | 0.0o | 16.0Ko |
| - OK shinken | 2.3Mo | 4.9Mo | 2.6Mo | 536.0Ko |
| - OK synchronizer | 2.3Mo | 6.8Mo | 4.5Mo | 0.0o |
| <hr/> | | | | |
| - OK (journal) | 200.0Mo | 200.0Mo | 0.0o | 0.0o |
| - OK TOTAL | 204.6Mo | 211.7Mo | 7.1Mo | 552.0Ko (0%) |
| --> No reorganisation need | | | | |

- Le script ne conseillera un compactage de la base de données que si le pourcentage d'espace perdu est > **50%**
 - Dans ce cas, il conseille de se référer à la documentation (voir la page [MongoDB - maitriser l'espace utilisé](#)).
- La ligne journal permet de voir la taille du journal de base de **Wired Tiger**, qui vaut tout le temps 200Mo.