

MongoDB - Méthode 2: compactage de la base in-place

Sommaire

[Compactage des bases](#)
 [Paramètres d'exécution](#)
 [Options génériques](#)
 [Options de connexion SSH](#)
 [Options d'authentification](#)
 [Options SSL/TLS](#)
 [Fonctionnement](#)

Compactage des bases

Le script suivant permet d'effectuer un compactage (*défragmentation*) des données stockées dans les différentes bases de données :

- [launch_mongo_compaction.py](#)

Il est important de noter que pendant que le script va fonctionner, la base sur laquelle il agit sera bloquée, et les opérations de Shinken ne pourront pas se faire.

Il faut arrêter Shinken avant de lancer ce script.



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

PS: Si vous avez besoin du script pour une version antérieure de Shinken, contacter votre support.

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 :

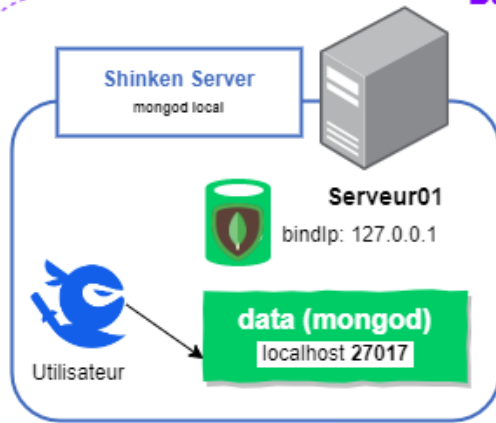
- La base de données MongoDB ne se trouve pas sur la machine qui exécute la commande.
- L'authentification par mot de passe à la base MongoDB est activée.
- Le port de MongoDB n'est pas celui par défaut (*défaut : 27017*).



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

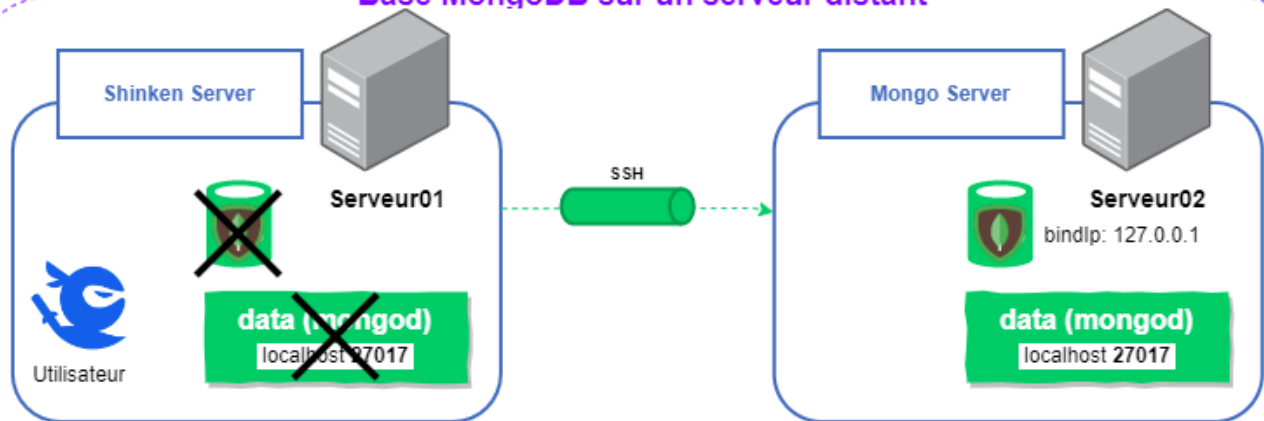


```
[root@serveur01 ~] shinken-commande --mongo-host 127.0.0.1 --mongo-port 27017 --mongo-database shinken
```

Option	Valeur par défaut	Description
<code>--mongo-host ARG</code>	localhost	Nom ou IP du serveur MongoDB.
<code>--mongo-port ARG</code>	27017	Port de la base MongoDB.
<code>--mongo-database ARG</code>	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

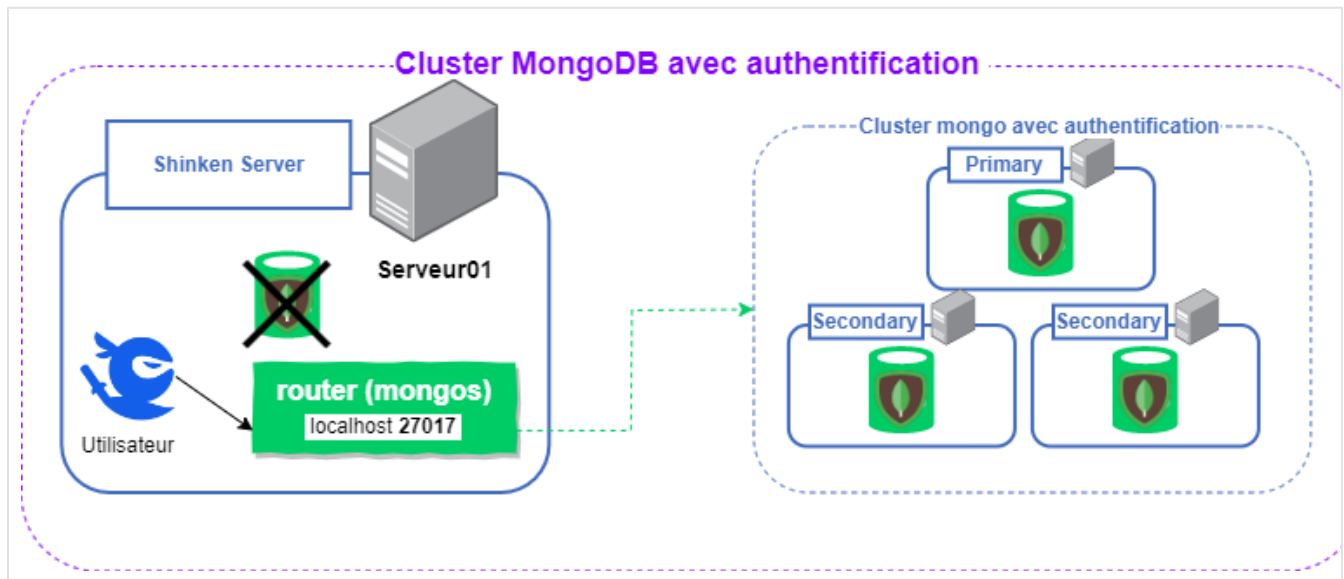
Base MongoDB sur un serveur distant



```
[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

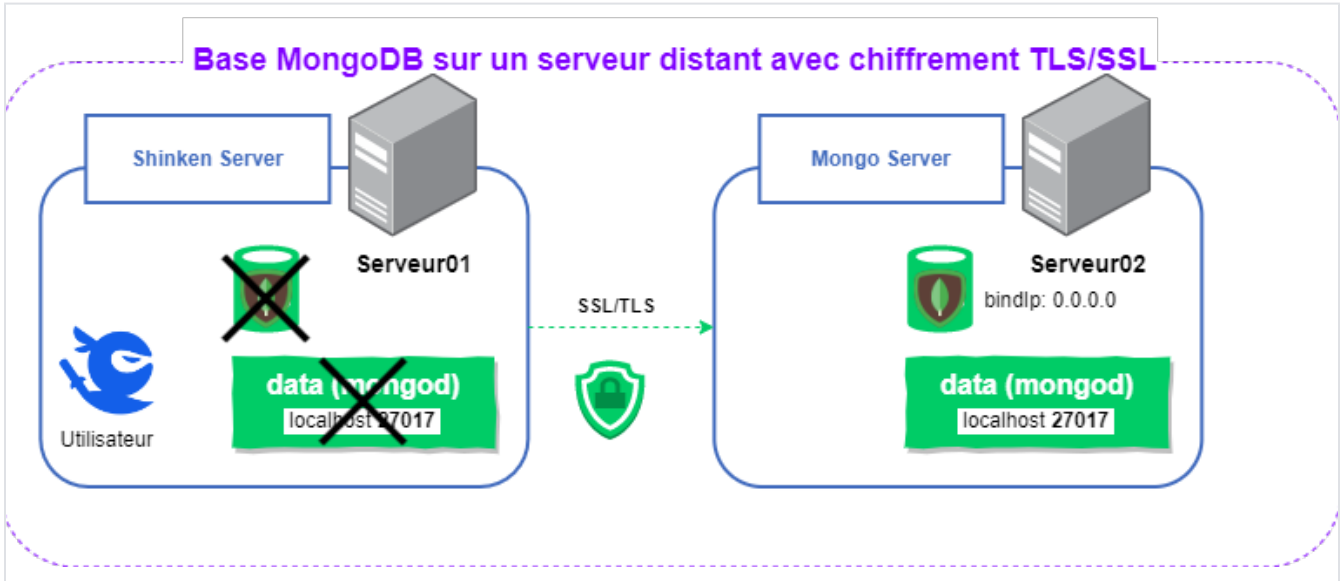


```
[root@serveur01 ~] shinken-command --mongo-host 127.0.0.1 --mongo-port 27017 --mongo-username shinken --mongo-password shinken
```

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

<pre>-- mongo -- password ARG</pre>	--- Mot de passe de l'utilisateur pour l'authentification avec mot de passe. À utiliser en complément de l'option <code>--mongo-username</code>. ✔ 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>). 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. 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>).
-------------------------------------	--

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> .

Fonctionnement



ATTENTION : lancer cette commande va bloquer la base pendant l'opération. La durée du traitement va dépendre de la vitesse du serveur, notamment de la vitesse de ses disques, car toutes les données vont être lues/réécrites. Il faut que Shinken soit arrêté pendant cette opération.

Le script va énumérer les bases présentes sur le serveur, et les compacter l'une après l'autre. Le temps passé sur chaque base sera affiché.

Voici un exemple d'exécution du script :

```
Mongodb storage engine: wiredTiger

- local      :  DONE (in 0.366s)
- shinken    :  DONE (in 99.386s)
- synchronizer : DONE (in 0.371s)
```



Cette procédure peut permettre de récupérer de l'espace disque mais elle n'est pas magique.

Il est important de limiter les données stockées par Shinken (*SLA, Événement*) pour que l'espace consommé soit plafonné. Pour cela, se référer à la page : [MongoDB - Méthode 1 : Ne garder que les données pertinentes](#)