

shinken-synchronizer-database-restore -- Restaurer une sauvegarde automatique de la base de données du Synchronizer

Sommaire

- Concept
 - Utilisation de la commande
 - Options de de paramétrage des sauvegardes à charger
 - Options de connexion à la base MongoDB
 - Options génériques
 - Options de connexion SSH
 - Options d'authentification
 - Options SSL/TLS
- Exemple

Concept

Shinken Entreprise V02.06.00 introduit un nouveau Module : `synchronizer-module-database-backup` permettant d'effectuer une sauvegarde automatique à intervalles réguliers de la base de données du Synchronizer (voir la page [Module synchronizer-module-database-backup](#)).

Cela permettra, en cas de mauvaise manipulation, de restaurer facilement une ancienne version grâce à la commande **shinken-synchronizer-database-restore**.



Cette commande ne permet de restaurer que la base de données MongoDB du Synchronizer ! Les fichiers de configuration, les logs, les SLA, les métriques et les données utilisateur ne sont pas sauvegardés par le module et ne sont donc pas inclus dans la restauration de cette commande.

Pour une restauration complète, voir la page [Shinken-backup et Shinken-restore, les commandes de sauvegarde et de restauration](#).

Utilisation de la commande

La commande **shinken-synchronizer-database-restore** permet de restaurer n'importe quelle version sauvegardée.

- Si plusieurs sauvegardes sont disponibles, elle propose la liste des sauvegardes disponibles avec un horodatage, triée de la plus ancienne à la plus récente.



Pour que la commande puisse s'exécuter, Shinken doit être arrêté.

Il est possible de choisir la sauvegarde à restaurer en saisissant son numéro.

- Afin que la restauration puisse être effectuée sans risque de corruption de données, la commande **shinken-synchronizer-database-restore** demande un accord pour arrêter Shinken durant la procédure de restauration.
- Une fois celle-ci effectuée, la commande demande un accord pour redémarrer Shinken.
- Lorsque de la restauration de la base de données du Synchronizer, il est possible de restaurer une base qui n'est pas compatible avec la version actuellement installée de Shinken (*c'est le cas si l'archive est antérieure à la mise à jour de Shinken*) .
 - Pour cette raison, la commande lance des actions automatiques afin d'assurer la compatibilité entre la base de données restaurée et la version actuellement installée de Shinken.

Options de de paramétrage des sauvegardes à charger

Option	Valeur par défaut	Description
--------	-------------------	-------------

-c	/etc/shinken/synchronizer.cfg	Chemin du fichier de configuration du Synchronizer.
-d	Le dossier spécifié dans le fichier de configuration du module "synchronizer-module-database-backup" (voir la page Module synchronizer-module-database-backup)	Chemin du dossier où se trouvent les archives de la base MongoDB du Synchronizer.
-f	---	Nom d'un backup à charger (sans l'extension).

Options de connexion à la base MongoDB



Cette commande récupère les paramètres de connexion à la base MongoDB depuis la configuration. Il est nécessaire d'utiliser les options de la ligne de commande que si les fichiers de configuration ne correspondent pas à la base MongoDB sur la quel la commande doit être exécutée (*migration de base, test sur une préprod ...*).

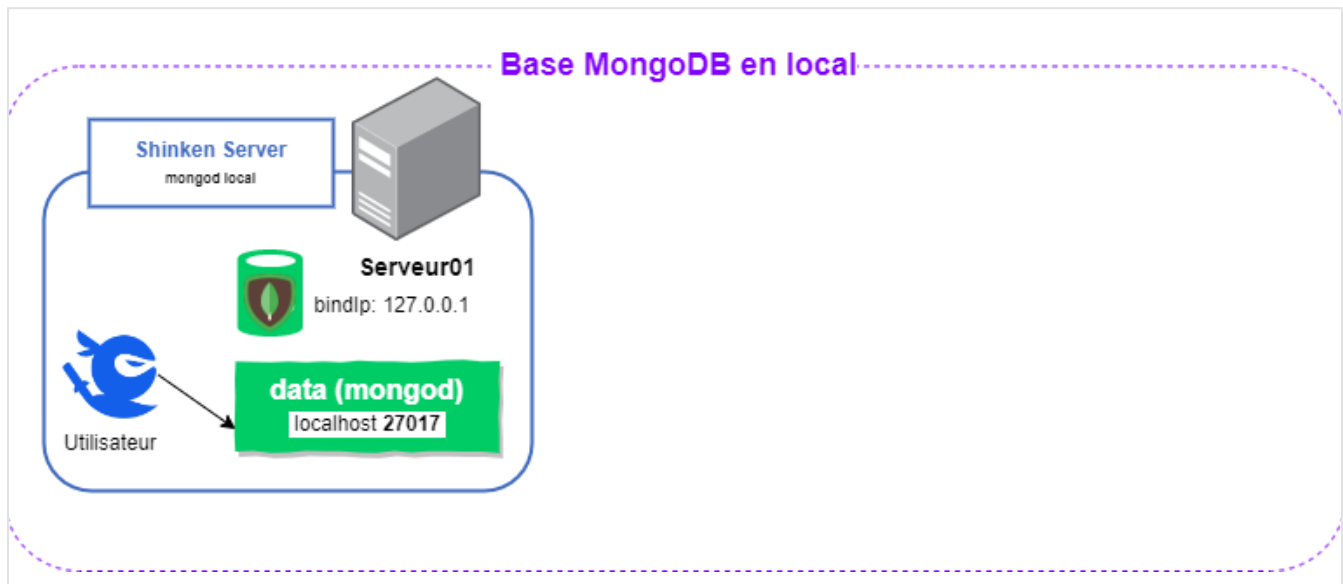
La commande dispose d'options de connexion à la base MongoDB qui peuvent être utilisés dans les cas suivants :

- 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

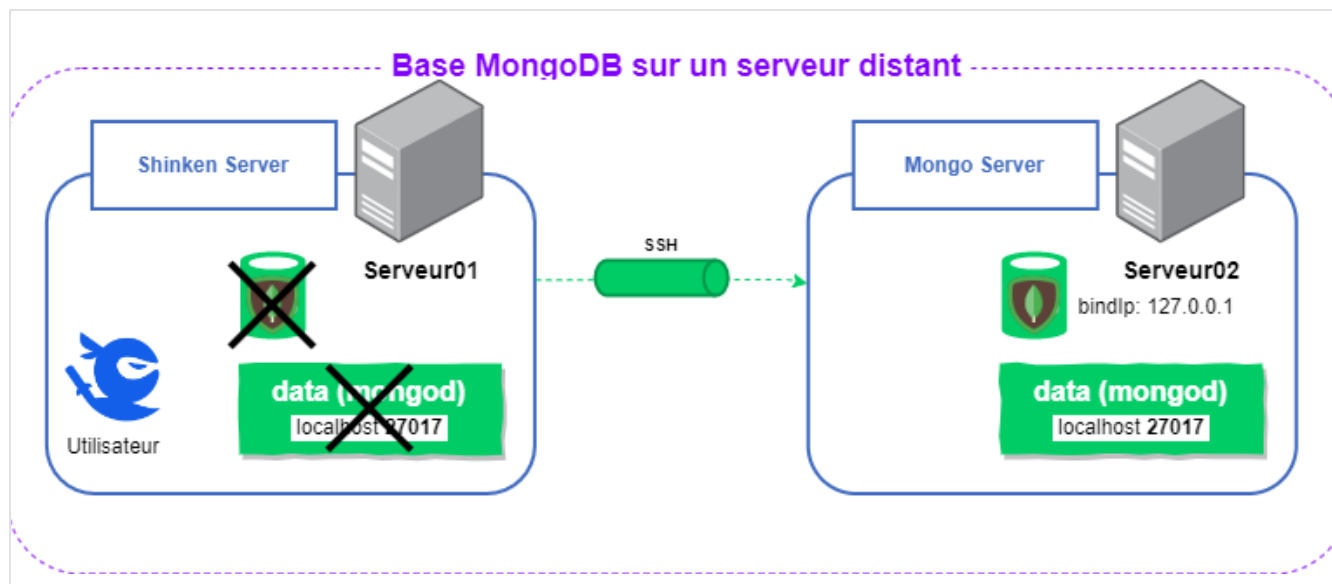


```
[root@serveur01 ~] shinken-commande --mongo-host 127.0.0.1 --mongo-port 27017 --mongo-database shinken
```

Option	Valeur par défaut	Description
<code>--mongo-host</code> <i>ARG</i>	localhost	Nom ou IP du serveur MongoDB.

<code>--mongo-port A</code> <i>RG</i>	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.

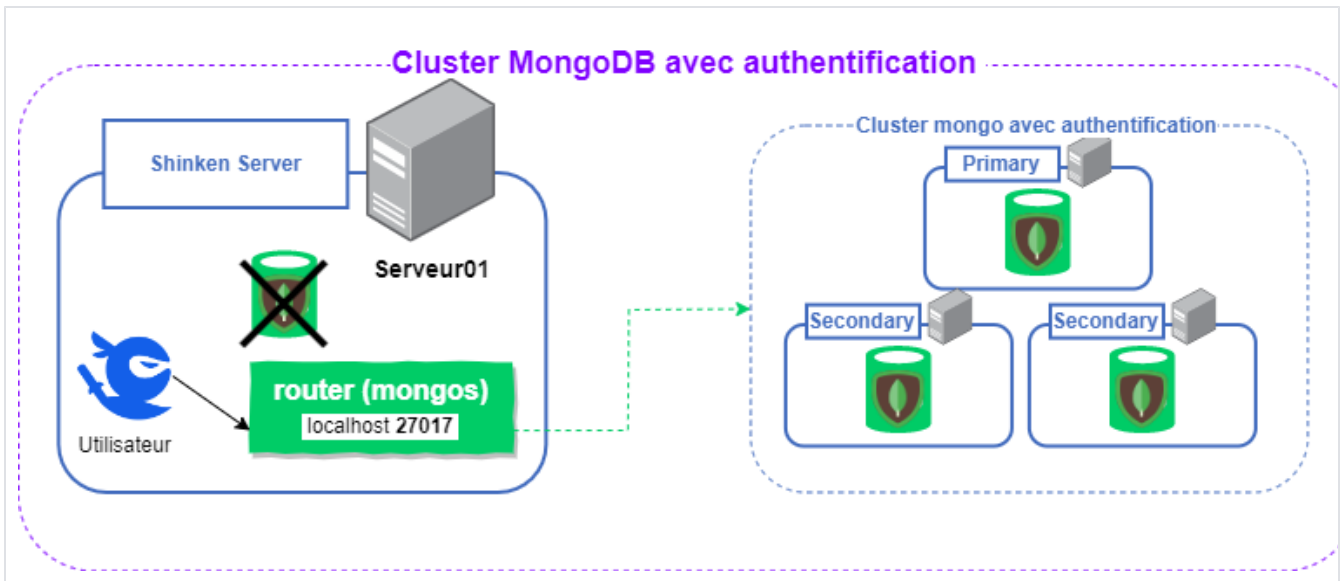
Options de connexion SSH



```
[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>	/var/lib/shinken/.ssh/id_rsa	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>	shinken	Utilisateur à utiliser pour la connexion SSH. À utiliser en complément de l'option <code>--mongo-use-ssh</code> .

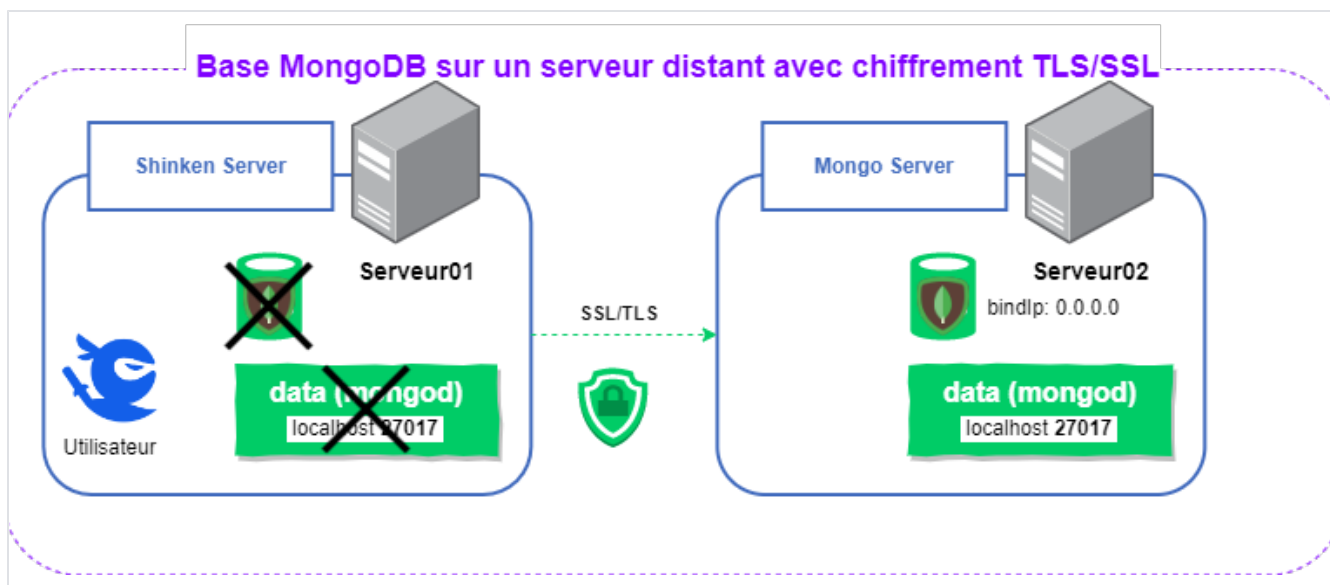
Options d'authentification



```
[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.
<code>--mongo-password ARG</code>	---	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 --mongo-password est utilisée, le mot de passe risque d'être visible dans l'historique des commandes (<i>via la commande history</i>). Pour éviter d'exposer le mot de passe, il est possible d'utiliser cette commande uniquement avec l'option --mongo-username. 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> .

Exemple

```

$ shinken-synchronizer-database-restore
Theses backup were found :
[ 0] 2024-04-24_13-07_synchronizer_localhost.tgz
[ 1] 2024-04-24_14-07_synchronizer_localhost.tgz
[ 2] 2024-04-24_15-04_synchronizer_localhost.tgz
[ 3] 2024-04-24_15-32_synchronizer_localhost.tgz
[ 4] 2024-04-24_15-46_synchronizer_localhost.tgz
[ 5] 2024-04-24_15-48_synchronizer_localhost.tgz
[ 6] 2024-04-24_15-50_synchronizer_localhost.tgz
[ 7] 2024-04-24_15-54_synchronizer_localhost.tgz
[ 8] 2024-04-24_15-55_synchronizer_localhost.tgz
[ 9] 2024-04-24_16-36_synchronizer_localhost.tgz
[10] 2024-04-25_08-18_synchronizer_localhost.tgz
[11] 2024-04-25_11-17_synchronizer_localhost.tgz
[12] 2024-04-25_12-12_synchronizer_localhost.tgz
Which backup do you want to restore ? [0-12] : 10
Will restore the 2024-04-25_08-18_synchronizer_localhost.tgz
2024-04-25_08-18_synchronizer_localhost.tgz
Will stop the Synchronizer in order to restore its database.
Do you want to stop the synchronizer now ? [y/N] : y
Stopping synchronizer:
[ (id:0) synchronizer-master ] : [ OK ]

Connexion to the MongoDB Synchronizer database
[2024-04-25 12:13:14] INFO : [ Shinken ] [ INITIALISATION ] [ MONGO ] Creating connection to database [synchronizer],
requested by [ synchronizer ]
[2024-04-25 12:13:14] INFO : [ Shinken ] [ INITIALISATION ] [ MONGO ] Try to open a Mongodb connection to
[ mongodb://localhost:27017/?fsync=false ] database [ synchronizer ]
[2024-04-25 12:13:14] INFO : [ Shinken ] [ INITIALISATION ] [ MONGO ] Mongo connection established in 0.01s
Extracting backup.
Dropping current Synchronizer Database.
Restoring the selected Synchronizer database.
Database restored. You can now restart your synchronizer
The backup [2024-04-25_08-18_synchronizer_localhost.tgz] was successfully restored.
Launch Sanatize :
fix_double_sync_keys : skip (unecessar
fix_remove_shinken_core : skip (unecessar
fix_remove_deprecated_check : skip (unecessar
fix_remove_undefined_aix_templates : skip (unecessar
fix_flapping_thresholds : skip (unecessar

```

Si le script est lancé avec l'option -f et que le backup indiqué n'existe pas, le script l'indiquera et proposera les backups disponibles, comme dans l'exemple ci-dessous :

```

$ shinken-synchronizer-database-restore -f this-backup-does-not-exists
Impossible to load the required backup
Theses backup were found :
[ 0] 2024-09-24_14-58_synchronizer_localhost.tgz
[ 1] 2024-09-24_15-49_synchronizer_localhost.tgz
[ 2] 2024-09-24_15-54_synchronizer_localhost.tgz
Which backup do you want to restore ? [0-2] :

```