

Sauvegarde et restauration manuelle de la rétention

Sommaire

Concept

Sauvegarde et restauration de la rétention avec le module `PickleRetentionFile`

Où trouver la rétention Pickle

Comment sauvegarder la rétention Pickle

Comment restaurer la rétention Pickle

Restoration des fichiers de rétention

Redémarrage du Scheduler

Sauvegarde et restauration de la rétention avec le module `MongodbRetention`

Où trouver la rétention Mongo

Comment sauvegarder la rétention Mongo

Avant de lancer la sauvegarde

Sauvegarder la rétention

Options de connexion à la base MongoDB

Exemple de sauvegarde

Sauvegarde de la rétention sur un MongoDB en local

Exemple de retour avec succès de la commande

Comment restaurer la rétention `MongodbRetention`

Avant de sauvegarder la rétention

Détails de la commande `mongorestore`

Options de connexion à la base MongoDB

Exemple de restauration

Restoration d'une rétention sur un MongoDB en local

Exemple de retour avec succès d'une commande de restauration

Après la restauration de la rétention

Concept

Shinken ne fournit pas de commandes natives pour sauvegarder ou restaurer une rétention. Cette page décrit le protocole à suivre pour effectuer ces opérations manuellement.

La procédure de sauvegarde et de restauration dépend du module de rétention utilisé par vos Scheduler :

- Le module `PickleRetentionFile` (voir la page [Module PickleRetentionFile \(Rétention par fichier plat \)](#)).
- Le module `MongodbRetention` (voir la page [Module MongodbRetention \(Rétention en base de données centralisée par royaume \)](#)).

Sauvegarde et restauration de la rétention avec le module `PickleRetentionFile`

Où trouver la rétention Pickle

Lorsque le module `PickleRetentionFile` est utilisé, la rétention est sauvegardée dans un fichier, situé par défaut dans le répertoire suivant : `/var/lib/shinken/`.

Ce chemin peut toutefois varier selon la configuration spécifique des modules `PickleRetentionFile` utilisés par les Scheduler (voir la page [Module PickleRetentionFile \(Rétention par fichier plat \)](#)).

Voici un exemple complet de nom de fichier de rétention Pickle :

```
/var/lib/shinken/retention.dat_--_realm--all_--_scheduler--scheduler-master_--_id--0.retention
```

Comment sauvegarder la rétention Pickle

Pour sauvegarder les fichiers de rétention, il suffit de les copier vers l'emplacement choisi en veillant à préserver leur intégrité et leurs permissions :

```
cp -a /var/lib/shinken/retention.dat_--_realm--all_--_scheduler--scheduler-master_--_id--0.retention /root/retention_save/
```

- `-a` : permet de préserver les droits et métadonnées du fichier

Comment restaurer la rétention Pickle

Restauration des fichiers de rétention

Pour restaurer un fichier de rétention Pickle, il faut le copier dans le dossier `/var/lib/shinken/` :

```
cp -a /root/retention_save/retention.dat--_realm--all--_scheduler--scheduler-master--_id--0.retention /var/lib/shinken/
```

Redémarrage du Scheduler

Après avoir restauré les fichiers de rétention, il faut redémarrer le **Scheduler** pour que les changements soient pris en compte :

```
service-shinken-scheduler restart
```

Sauvegarde et restauration de la rétention avec le module MongoDBRetention

Où trouver la rétention Mongo

Lorsque le module **MongodbRetention** est utilisé, la rétention est sauvegardée dans la base **shinken** de MongoDB (voir la page [Module MongodbRetention \(Rétention en base de données centralisée par royaume \)](#)), au sein des collections **retention_hosts_raw** et **retention_services_raw**.



Il est impératif de sauvegarder et de restaurer **les deux collections** pour garantir l'intégrité de la rétention.

Comment sauvegarder la rétention Mongo

Avant de lancer la sauvegarde

Avant de procéder à la sauvegarde de la rétention, il est conseillé de redémarrer le Scheduler afin de forcer la mise à jour de la rétention et d'obtenir ainsi des données plus fraîches :

```
service-shinken-scheduler restart
```

Sauvegarder la rétention

Pour enregistrer la rétention, il faut lancer les deux commandes suivantes :

```
mongodump -d shinken -c retention_hosts_raw -o /root/retention_mongo/ ( + MONGO_DB_CONNECTION_OPTIONS )
```

```
mongodump -d shinken -c retention_services_raw -o /root/retention_mongo/ ( + MONGO_DB_CONNECTION_OPTIONS )
```

- **shinken** : nom de la base définie dans la configuration du module (voir la page [Module MongodbRetention \(Rétention en base de données centralisée par royaume \)](#)).
- **/root/retention_mongo/** : dossier où seront sauvegardées les rétentions. Le chemin peut être absolu ou relatif (dans ce dernier cas, le dossier sera créé à l'emplacement où la commande est exécutée).
- **retention_hosts_raw / retention_services_raw** : noms des collections où se trouvent les données de rétentions. Ne pas les modifier dans la commande.
- **MONGO_DB_CONNECTION_OPTIONS** : options de connexion à MongoDB. Ces options sont nécessaires si l'authentification, le chiffrement ou des paramètres spécifiques sont activés sur la base (voir le chapitre [Options de connexion à la base MongoDB](#)).



Dans le cas d'un cluster MongoDB, il faut s'assurer de communiquer avec un **mongos**.


La commande crée les fichiers suivants :

- retention_hosts_raw.bson
- retention_hosts_raw.metadata.json
- retention_services_raw.bson
- retention_services_raw.metadata.json

Options de connexion à la base MongoDB

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*).
- Dans le cas d'un cluster MongoDB, il n'y a pas de mongos en local pour router les requêtes vers la base de données.

Option	Valeur par défaut	Description
<code>--host ARG</code>	localhost	Nom ou IP du serveur MongoDB.
<code>--port ARG</code>	27017	Port de la base MongoDB.
<code>--username [-u] ARG</code>	--	Utilisateur pour l'authentification avec mot de passe.
<code>--password [-p] 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> . <div style="border: 1px solid green; padding: 10px; margin-top: 10px;"><p> Si l'option <code>--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>-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>
<code>--authenticationDatabase ARG</code>	--	Base de données où l'utilisateur utilisé pour l'authentification avec mot de passe a été créé. À utiliser en complément de l'option <code>--mongo-username</code> .
<code>--ssl</code>	--	Active SSL/TLS pour les communications avec la base MongoDB.
<code>--sslCAFile 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>--ssl</code> .
<code>--sslPEMKeyFile ARG</code>	--	Chemin vers le fichier contenant le certificat SSL du client. À utiliser en complément de l'option <code>--ssl</code> .

<code>-- sslPEMK eyPassw ord ARG</code>	--	Mot de passe du certificat SSL du client. À utiliser en complément de l'option <code>--ssl</code> .
<code>-- sslCRLFil e 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>--ssl</code> .
<code>-- sslAllowIn validHost names</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>--ssl</code> .
<code>-- sslAllowIn validCertif icates</code>	--	Accepter le certificat SSL de MongoDB même s'il est invalide, par exemple expiré. À utiliser en complément de l'option <code>--ssl</code> .

Exemple de sauvegarde

Sauvegarde de la rétention sur un MongoDB en local

Pour sauvegarder la rétention depuis un MongoDB sur le même serveur où les commandes sont lancées :

```
mongodump -d shinken -c retention_hosts_raw -o /root/retention_mongo/
mongodump -d shinken -c retention_services_raw -o /root/retention_mongo/
```



Il faut faire la sauvegarde en local sur la machine. Pour des raisons de sécurité, MongoDB écoute que sur l'interface locale (*localhost*).

Exemple de retour avec succès de la commande

```
2025-06-04T03:58:27.264-0400   writing shinken.retention_services_raw to /root/retention_mongo/shinken
/retention_services_raw.bson
2025-06-04T03:58:27.269-0400   writing shinken.retention_services_raw metadata to /root/retention_mongo
/shinken/retention_services_raw.metadata.json
2025-06-04T03:58:27.272-0400   done dumping shinken.retention_services_raw (32 documents)
```

Si la dernière ligne du retour de la commande, il y a écrit "(0 documents)" :

- La première sauvegarde de la rétention n'a pas encore eu lieu.
 - Attendre la première sauvegarde de la rétention (*par défaut 60 minutes*, voir la page [Fichier de configuration \(shinken.cfg \)](#)) ou redémarrer le Scheduler pour la forcer.
- Les paramètres fournis sont incorrects. Vérifier le nom de la base, des collections.

Comment restaurer la rétention MongoDBRetention

Avant de sauvegarder la rétention

Avant d'effectuer la restauration de la rétention, il est conseillé d'éteindre le Scheduler :

```
service-shinken-scheduler stop
```

Détails de la commande mongorestore

Pour restaurer la rétention, lancer la commande :


```
mongorestore --drop /root/retention_mongo/ ( + MONGO_DB_CONNECTION_OPTIONS )
```

- **--drop** : Permet de supprimer les collections de rétention présentes dans la base afin d'éviter toute incohérence.
- **/root/retention_mongo/** : Dossier contenant la sauvegarde de la rétention, créé par la commande mongodump (voir le chapitre [mongodump](#)).
- **MONGO_DB_CONNECTION_OPTIONS** : Options de connexion à MongoDB. Nécessaire si l'authentification ou le chiffrement de la base est activé, ou si la base n'utilise pas les paramètres par défauts.

Options de connexion à la base MongoDB

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*).
- Dans le cas d'un cluster MongoDB, il n'y a pas de mongos en local pour router les requêtes vers la base de données.

Option	Valeur par défaut	Description
<code>--host ARG</code>	localhost	Nom ou IP du serveur MongoDB.
<code>--port ARG</code>	27017	Port de la base MongoDB.
<code>--username [-u] ARG</code>	--	Utilisateur pour l'authentification avec mot de passe.
<code>--password [-p] 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> . <div style="border: 1px solid green; padding: 5px; margin-top: 10px;"> Si l'option <code>--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>-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>).</div>
<code>--authenticationDatabase ARG</code>	--	Base de données où l'utilisateur utilisé pour l'authentification avec mot de passe a été créé. À utiliser en complément de l'option <code>--mongo-username</code> .

<code>--ssl</code>	--	Active SSL/TLS pour les communications avec la base MongoDB.
<code>--sslCAFile 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>--ssl</code> .
<code>--sslPEMKeyFile ARG</code>	--	Chemin vers le fichier contenant le certificat SSL du client. À utiliser en complément de l'option <code>--ssl</code> .
<code>--sslPEMKeyPassword ARG</code>	--	Mot de passe du certificat SSL du client. À utiliser en complément de l'option <code>--ssl</code> .
<code>--sslCRLFile 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>--ssl</code> .
<code>--sslAllowInvalidHostnames</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>--ssl</code> .
<code>--sslAllowInvalidCertificates</code>	--	Accepter le certificat SSL de MongoDB même s'il est invalide, par exemple expiré. À utiliser en complément de l'option <code>--ssl</code> .



Dans le cas d'un cluster MongoDB, il faut s'assurer de communiquer avec un **mongos**.

Exemple de restauration

Restauration d'une rétention sur un MongoDB en local

Cette commande va restaurer dans un MongoDB local les données contenues dans le dossier **/root/retention_mongo/** :

```
mongorestore --drop /root/retention_mongo/
```



Il faut faire la restauration en local sur la machine. Pour des raisons de sécurité, MongoDB écoute que sur l'interface locale (*localhost*).

Exemple de retour avec succès d'une commande de restauration

```
2025-06-04T04:22:19.606-0400    building a list of dbs and collections to restore from retention_mongo/ dir
2025-06-04T04:22:19.611-0400    reading metadata file from retention_mongo/shinken/retention_services_raw.
metadata.json
2025-06-04T04:22:19.611-0400    restoring shinken.retention_services_raw from file retention_mongo/shinken
/retention_services_raw.bson
2025-06-04T04:22:19.661-0400    reading metadata file from retention_mongo/shinken/retention_hosts_raw.
metadata.json
2025-06-04T04:22:19.661-0400    restoring shinken.retention_hosts_raw from file retention_mongo/shinken
/retention_hosts_raw.bson
2025-06-04T04:22:19.666-0400    restoring indexes for collection shinken.retention_services_raw from metadata
2025-06-04T04:22:19.700-0400    finished restoring shinken.retention_services_raw (32 documents)
2025-06-04T04:22:19.701-0400    restoring indexes for collection shinken.retention_hosts_raw from metadata
2025-06-04T04:22:19.702-0400    finished restoring shinken.retention_hosts_raw (2 documents)
```

Si la dernière ligne du retour de la commande, il y a écrit "(0 documents)" :

- La première sauvegarde de la rétention n'a pas encore eu lieu.
 - Attendre la première sauvegarde de la rétention (*par défaut 60 minutes, voir la page [Fichier de configuration \(shinken.cfg \)](#)*) ou redémarrer le Scheduler pour la forcer.
- Les paramètres fournis sont incorrects.
 - Vérifier le nom de la base, des collections.
- Il y a eu une erreur lors de la sauvegarde précédente.

Après la restauration de la rétention

Une fois la restauration terminée, il faut redémarrer le Scheduler :

```
service-shinken-scheduler restart
```