

# 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

        Restauration 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

            Restauration d'une rétention sur un MongoDB en local

            Exemple de retour avec succès d'une commande de restauration

        Après la restauration de la rétention

## Description

La méthode pour sauvegarder et restaurer manuellement la rétention dépend du type de module de rétention utilisé par vos Schedulers :

- Le module `PickleRetentionFile` ( voir le chapitre [Sauvegarde et restauration de la rétention avec le module `PickleRetentionFile`](#) ) ;
- Le module `MongodbRetention` ( voir le chapitre [Sauvegarde et restauration de la rétention avec le module `MongodbRetention`](#) ) ;



Vous trouverez plus d'information sur ces modules sur les pages décrivant leur mise en place :

- [Module `PickleRetentionFile` \( Rétention par fichier plat \)](#)
- [Module `MongodbRetention` \( Rétention en base de donnée centralisée par royaume \)](#)

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

### Où trouver la rétention Pickle

Lorsque le module `PickleRetentionFile` est utilisé, la sauvegarde de la rétention se fait dans un fichier situé par défaut au chemin suivant : `/var/lib/shinken/`

Il est possible que ce chemin ait changé en fonction de la configuration des modules `PickleRetentionFile` utilisés par vos Schedulers.



Pour plus d'informations sur la configuration des modules `PickleRetentionFile`, référez-vous à cette 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
```



Pour plus d'informations, référez-vous au chapitre détaillant le chemin des fichiers de rétention Pickle de la page expliquant le module `PickleRetentionFile` ( voir dans la page de configuration du Module `PickleRetentionFile` le chapitre [Détails de le chemin des données de rétention](#) )

## Comment sauvegarder la rétention Pickle

Pour sauvegarder ces fichiers de rétention, il suffit de les copier à l'endroit souhaité avec la commande `cp <source> <destination>` :

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

Cette commande va copier le fichier de rétention `retention.dat_--_realm--all_--_scheduler--scheduler-master_--_id--0.retention` dans le dossier nommé `/root/retention_save/`



- Le dossier de destination `/root/retention_save/` doit exister avant de lancer la commande.
- Le dossier de destination `/root/retention_save/` n'est qu'un exemple, choisissez où bon vous semble.

## Comment restaurer la rétention Pickle

### Restauration des fichiers de rétention

Pour restaurer un fichier de rétention Pickle, il suffit de le remettre à l'endroit depuis lequel ils ont été copiés : `/var/lib/shinken/`

Si les fichiers ont été copiés dans un dossier nommé `/root/retention_save/`, alors la commande pour restaurer la rétention devrait ressembler à ça :

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



Attention, la commande `cp` **écrase la destination** !

Avant de restaurer les fichiers Pickle, assurez-vous qu'il n'y a pas de fichier du même nom dans `/var/lib/shinken/`, ou assurez-vous de les sauvegarder si besoin.

### Redémarrage du Scheduler

Après avoir restauré les fichiers de rétention, il faut redémarrer le **Scheduler** pour que les changements fassent effet :

```
service shinken-scheduler restart
```

## Sauvegarde et restauration de la rétention avec le module MongoDBRetention

### Où se trouve la rétention dans Mongo

Avec le module MongoDBRetention, la rétention se trouve dans la base `shinken` ( *par défaut* ) dans les collections `retention_hosts_raw` et `retention_services_raw`.



Pensez à vérifier que vous n'avez pas changé le nom de la base par défaut ( `shinken` ), dans le cas contraire, il faudra changer le nom de la base dans la commande ( *paramètre "-d"* )

## Comment sauvegarder la rétention MongoDBRetention

### Avant de lancer la sauvegarde

Avant de sauvegarder la rétention, il est conseillé ( *si possible* ) de redémarrer le Scheduler afin de forcer la rétention, et donc d'avoir des données plus fraîches :

```
service shinken-scheduler restart
```

## Détails de la commande mongodump

Pour enregistrer les données de ces collections, on utilise la commande **mongodump** :

```
mongodump -d DATABASE_NAME -c COLLECTION_NAME -h HOST_NAME --port=HTTP_PORT --ssl -o OUTPUT_NAME
```

Explication des paramètres :

Paramètre	Obligatoire	Exemple	Description
-d DATABASE_NAME	Oui	shinken	Nom de la base Mongo
-c COLLECTION_NAME	Oui	retention_hosts_raw  retention_services_raw	Nom de la collection Mongo Les deux collections possible sont : <ul style="list-style-type: none"><li>retention_hosts_raw : pour les hôtes et les clusters</li><li>retention_services_raw : pour les checks</li></ul>
-h HOST_NAME	Non	192.168.1.25	Adresse IP ou nom de l'hôte où se trouve la base MongoDB utilisée par le Scheduler pour sauvegarder sa rétention. Non nécessaire si la commande est lancée sur le même serveur.
--port=HTTP_PORT	Non	27018	Port HTTP de la <b>base MongoDB</b> ( <i>et non du serveur</i> ). Par défaut le port utilisé est <b>27017</b> .  Ce port peut être trouvé par défaut dans le fichier <code>/etc/mongod.conf</code> ( <i>ou /etc/mongos.conf si vous utilisez MongoDB en cluster</i> ).  Si vous avez changé le port de MongoDB ( <i>par défaut 27017</i> ) il faut le préciser via ce paramètre, <b>même en local</b> .
--ssl	Non	-	Si votre base MongoDB utilise le SSL, il faut ajouter cette option qui ne prend pas d'arguments.
-o OUTPUT_NAME	Non	retention_hosts_11_10_2021	Nom du fichier de sortie de la commande. Ce paramètre est optionnel, mais nous vous recommandons de donner un nom parlant au dossier créé par <b>mongodump</b> . Si ce paramètre n'est pas renseigné, alors le dossier sera nommé <b>dump</b> .



Si vous utilisez MongoDB en mode cluster, il faut s'assurer de communiquer avec le bon service. Cette explication se trouve dans le chapitre expliquant le cas particulier du cluster MongoDB ( voir le chapitre [Cas particulier de l'utilisation d'un cluster MongoDB](#) ).



- Pour savoir quels paramètres utiliser, vous pouvez vous référer au fichier de configuration de votre Scheduler, par défaut ici : `/etc/shinken/schedulers/scheduler-master.cfg`
- Il est possible d'utiliser plusieurs fois le même OUTPUT\_NAME, les données seront alors toutes sauvegardées dans le même dossier et donc peuvent être toutes restaurées avec qu'une seule commande **mongorestore**.

Ces deux commandes vont créer deux dossiers à l'emplacement `/root/retention_mongo/` ( *assurez-vous que ce chemin existe* ) nommé **retention\_hosts\_11\_10\_2021** et **retention\_services\_11\_10\_2021** contenant des fichiers `.json` et `.bson` qui contiennent les données des collections correspondantes.

## Comment faire la sauvegarde

Il faut que vous utilisiez la commande mongodump en fonction de votre situation :

- Vous êtes sur la machine qui héberge les données de la rétention => Il s'agit une sauvegarde en local
- Sinon, vous pouvez faire la sauvegarde de la machine où vous êtes en ciblant le serveur avec les données.

La configuration de votre module de rétention vous donnera l'information du serveur MongoDB où sont les données de rétention.

### Sauvegarde de la rétention sur une MongoDB en local

Pour sauvegarder la rétention dans deux dossiers **retention\_hosts\_11\_10\_2021** et **retention\_services\_11\_10\_2021** depuis un MongoDB sur le même serveur où ces commandes sont lancées :

```
mongodump -d shinken -c retention_hosts_raw -o /root/retention_mongo/retention_hosts_11_10_2021
mongodump -d shinken -c retention_services_raw -o /root/retention_mongo/retention_services_11_10_2021
```

### Sauvegarde de la rétention sur une MongoDB sur un autre serveur

Pour sauvegarder la rétention dans deux dossiers **retention\_hosts\_11\_10\_2021** et **retention\_services\_11\_10\_2021** depuis un MongoDB sur un serveur hébergé sur un serveur nommé **host\_bordeaux** et où MongoDB utilise le port **27018**.

```
mongodump -d shinken -c retention_hosts_raw -h host_bordeaux --port=27018 -o /root/retention_mongo/retention_hosts_11_10_2021
mongodump -d shinken -c retention_services_raw -h host_bordeaux --port=27018 -o /root/retention_mongo/retention_services_11_10_2021
```

### Sauvegarde de la rétention sur une MongoDB sur un autre serveur qui utilise le SSL

Pour sauvegarder la rétention dans deux dossiers **retention\_hosts\_11\_10\_2021** et **retention\_services\_11\_10\_2021** depuis un MongoDB sur un serveur hébergé sur un serveur nommé **host\_bordeaux**, où MongoDB utilise le port **27017** (*port par défaut*) et utilisant le SSL :

```
mongodump -d shinken -c retention_hosts_raw -h host_bordeaux --ssl -o /root/retention_mongo/retention_hosts_11_10_2021
mongodump -d shinken -c retention_services_raw -h host_bordeaux --ssl -o /root/retention_mongo/retention_services_11_10_2021
```

### Exemple de retour avec succès de la commande

```
2021-10-12T11:09:12.823+0200   writing shinken.retention_services_raw to /root/retention_hosts_11_10_2021
/root/shinken/retention_services_raw.bson
2021-10-12T11:09:12.823+0200   writing shinken.retention_services_raw metadata to /root
/root/retention_hosts_11_10_2021/shinken/retention_services_raw.metadata.json
2021-10-12T11:09:12.825+0200   the collection shinken.retention_services_raw appears to have been dropped
after the dump started
2021-10-12T11:09:12.825+0200   done dumping shinken.retention_services_raw (45 documents)
```



Si sur la dernière ligne du retour de la commande, il y a écrit "(0 documents)", alors il y a peut-être un problème quelque part, comme le nom de la collection qui n'est pas correcte.

## Comment restaurer la rétention MongoDB Retention

### 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

La restauration des données issues de la commande **mongodump** se fait avec la commande **mongorestore** :

```
mongorestore --drop --ssl -h HOST_NAME --port=HTTP_PORT RETENTION_FOLDER_PATH
```

Paramètre	Obligatoire	Exemple	Description
<code>--drop</code>	<b>Non mais conseillé</b>	-	<p>Sans ce paramètre, MongoDB va <b>tenter</b> de garder les collections présentes actuellement dans la base, ce qui peut résulter en des mélanges de données non souhaitées. Pour retrouver la rétention au même état qu'au moment de sa sauvegarde, il faut utiliser ce paramètre.</p> <div style="border: 1px solid red; padding: 5px; margin-top: 10px;">  Ce paramètre va effacer la rétention actuellement présente en base. </div>
<code>--ssl</code>	<b>Non</b>	-	Si votre base MongoDB utilise le SSL, il faut ajouter cette option qui ne prend pas d'arguments.
<code>-h</code> HOST_NAME	<b>Non</b>	192.168.1.25	Adresse IP ou nom de l'hôte où se trouve la base MongoDB utilisée par le Scheduler pour sauvegarder sa rétention. Non nécessaire si la commande est lancée sur le même serveur.
<code>--port=HTTP_PORT</code>	<b>Non</b>	27018	<p>Port HTTP de la <b>base MongoDB</b> ( <i>et non du serveur</i> ). Par défaut le port utilisé est <b>27017</b>.</p> <p>Ce port peut être trouvé par défaut dans le fichier <code>/etc/mongod.conf</code> ( <i>ou /etc/mongos.conf si vous utilisez MongoDB en cluster</i> ).</p> <p>Si vous avez changé le port de MongoDB ( <i>par défaut 27017</i> ) il faut le préciser via ce paramètre, <b>même en local</b>.</p>
RETENTION_FOLDER_PATH	<b>Oui</b>	<code>/root/retention_hosts_11_10_2021/</code>	<p>Chemin vers le dossier contenant la sauvegarde de la rétention qui a été créé avec la commande <code>mongodump</code> ( <i>pour plus d'explication voir le chapitre <a href="#">mongodump</a></i> ).</p> <p>Le dossier doit être sur le même serveur que celle où la commande est lancée.</p>



Si vous utilisez MongoDB en mode cluster, il faut s'assurer de communiquer avec le bon service. Cette explication se trouve dans le chapitre expliquant le cas particulier du cluster MongoDB ( voir le chapitre [Cas particulier de l'utilisation d'un cluster MongoDB](#) ).

## Comment faire la restauration

Il faut que vous utilisiez la commande `mongorestore` en fonction de votre situation :

- Vous êtes sur la machine qui héberge les données de la rétention => Il s'agit une restauration local
- Sinon, vous pouvez faire la restauration depuis la machine où vous êtes en ciblant le serveur où mettre les données.

La configuration de votre module de rétention vous donnera l'information du serveur MongoDB où doivent être les données de rétention.

### 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_hosts_11_10_2021/`

```
mongorestore --drop /root/retention_hosts_11_10_2021/
```

### Restauration d'une rétention sur un MongoDB sur un autre serveur

Cette commande va restaurer dans un MongoDB sur un serveur nommé `host_bordeaux` où MongoDB utilise le port `27018` les données contenues dans le dossier `/root/retention_hosts_11_10_2021/`

```
mongorestore --drop -h host_bordeaux --port=27018 /root/retention_hosts_11_10_2021/
```

### Restauration d'une rétention sur un MongoDB sur un autre serveur qui utilise le SSL

Cette commande va restaurer dans un MongoDB sur un serveur nommé `host_bordeaux` avec le port `27018` et utilisant le SSL, les données contenues dans le dossier `/root/retention_hosts_11_10_2021/`

```
mongorestore --drop --ssl -h host_bordeaux --port=27018 /root/retention_hosts_11_10_2021/
```

## Exemple de retour avec succès d'une commande de restauration

```
2021-10-12T11:11:01.024+0200    building a list of dbs and collections to restore from /root
/retention_hosts_11_10_2021/ dir
2021-10-12T11:11:01.026+0200    reading metadata file from /root/retention_hosts_11_10_2021/shinken
/retention_services_raw.metadata.json
2021-10-12T11:11:01.026+0200    restoring shinken.retention_services_raw from file /root
/retention_hosts_11_10_2021/shinken/retention_services_raw.bson
2021-10-12T11:11:01.027+0200    finished restoring shinken.retention_services_raw (45 documents)
2021-10-12T11:11:01.027+0200    done
```



Si sur l'avant-dernière ligne du retour de la commande, il y a écrit "(0 documents)", alors il y a peut-être un problème quelque part, comme la commande **mongodump** qui s'est mal passée ( *nom de collection incorrect par exemple* )

## Après la restauration de la rétention

Une fois la restauration terminée, vous pouvez redémarrer le Scheduler :

```
service shinken-scheduler start
```

## Cas particulier de l'utilisation d'un cluster MongoDB

Si vous utilisez MongoDB en mode cluster, les commandes restent les mêmes, à la nuance près qu'il faut préciser le port du bon service et il faut s'assurer d'avoir fait la sauvegarde de tous les replicaset.

- Pour sauvegarder l'entièreté d'un replicaset, il faut s'assurer de lancer les commandes sur le service **mongos** et non **mongod**.
- Pour utiliser un service plutôt que l'autre, il suffit de trouver et utiliser son port à fournir aux commandes.



Pour plus d'informations sur les clusters et les termes de MongoDB liés à ce sujet, référez-vous à la page d'explication sur la mise en place d'un cluster MongoDB ( voir la page [Haute disponibilité de la base MongoDB \(mise en place d'un cluster\)](#) )

## Vérifier la présence du service mongos

Pour vérifier si un mongos est sur un serveur et qu'il est opérationnel, il est possible de vérifier en demandant le statut du service **mongos** :

```
service mongos status
```

Si un processus **mongos** tourne sur le serveur sur lequel la commande a été lancée, alors le retour de la commande devrait ressembler à ça :

```
mongos.service - SYSV: Mongo is a scalable, document-oriented database.
Loaded: loaded (/etc/rc.d/init.d/mongos; bad; vendor preset: disabled)
Active: active (running) since Tue 2021-10-19 11:04:58 CEST; 53min ago
Docs: man:systemd-sysv-generator(8)
Process: 2805 ExecStart=/etc/rc.d/init.d/mongos start (code=exited, status=0/SUCCESS)
CGroup: /system.slice/mongos.service
        2820 /usr/bin/mongos -f /etc/mongos.conf
```



Si la commande précédente échoue, il est possible que le processus **mongos** se trouve sur un autre serveur.

## Trouver le port du service mongos

Le port se trouve dans le fichier de configuration de **mongos** ( *par défaut dans /etc/mongos.conf* ) :

```
...  
  
# network interfaces  
net:  
  port: 27018  
  bindIp: 192.168.1.100  
  
sharding:  
  configDB: primary:27019, secondary1:27019, secondary2:27019  
  
...
```

Le port souhaité est celui indiqué dans la section "net", dans notre exemple, ce sera le port du service **mongos** est **27018**.

### Exemples de restauration et de sauvegarde

Après avoir trouvé le port du bon service, disons **27018**, il faut le renseigner aux commandes de sauvegarde et de restauration :

#### Exemple de sauvegarde

```
mongodump --port=27018 -d shinken -c retention_hosts_raw -o /root/retention_mongo/retention_hosts_11_10_2021  
mongodump --port=27018 -d shinken -c retention_services_raw -o /root/retention_mongo/  
/retention_services_11_10_2021
```



Pour plus de détails sur la commande mongodump, référez-vous au chapitre : [Détails de la commande mongodump](#)

#### Exemple de restauration

```
mongorestore --port=27018 --drop /root/retention_mongo/retention_hosts_11_10_2021  
mongorestore --port=27018 --drop /root/retention_mongo/retention_services_11_10_2021
```



Pour plus de détails sur la commande mongorestore, référez-vous au chapitre : [Détails de la commande mongorestore](#)

### Faire la sauvegarde et restauration de tous les replicaset

Pour sauvegarder et restaurer l'entièreté d'un cluster MongoDB, il faut faire le tour de tous les replicaset.

Il peut y avoir plusieurs services **mongos** chargés des mêmes replicaset, il n'est pas nécessaire de faire une sauvegarde / restauration par mongos, mais une par replicaset.

Pour vérifier quel replicaset un service mongos est chargé, vous pouvez utiliser cette commande :

```
mongo --quiet --port MONGOS_PORT --eval "printjson(rs.status().set)"
```



Pensez à remplacer MONGOS\_PORT par le port du service mongos, vous pouvez vous référez au chapitre sur comment trouver le port du service mongos ( voir le chapitre [Trouver le port du service mongos](#) )

Cette commande retourne le nom du replicaset gérée par le service ayant le port 27018.

Exemple de retour :

```
$ mongo --quiet --port 27018 --eval "printjson(rs.status().set)"  
"replicaset-name"
```



Pour plus d'informations sur la récupération d'informations d'un replicaset, référez-vous à l'étape 4 de la procédure de configuration de la page sur la mise en place d'un cluster MongoDB ( voir le chapitre [Etape 4: Déclaration du replicaset dans Mongo](#) )

Une fois vos replicaset identifiés, vous pouvez lancer les commande de sauvegarde et de restauration sur **un** mongos par replicaset.

Les détails de ces commandes sont disponibles dans ces chapitres :

- [Détails de la commande mongodump](#)
- [Détails de la commande mongorestore](#)