# MongoDB - activation de l'authentification par mot de passe

**Sommaire**



## Concept

Pour garantir que seules les instances de Shinken puissent accéder aux données de la base, il est possible d'activer l'authentification par mot de passe au niveau de la base de données.

Il sera alors nécessaire de spécifier les identifiants à utiliser par Shinken dans les fichiers de configuration des démons et des modules se connectant à MongoDB.

Ces identifiants devront être fournis lors de l'exécution de toute commande Shinken nécessitant un accès à la base de données.

La procédure décrite permet de minimiser au maximum le temps d'interruption de service. Ce temps d'interruption correspondra uniquement au temps nécessaire pour redémarrer le cluster MongoDB et Shinken.

Les étapes du protocole sont les suivantes :

1. Créer l'utilisateur dans la base MongoDB.
2. Exiger l'authentification par mot de passe dans MongoDB.
3. Configurer Shinken pour s'authentifier avec l'utilisateur auprès de MongoDB.
4. Redémarrer MongoDB et Shinken.
5. Mettre à jour la supervision ( *sup de sup* ) des Brokers et du Synchronizer dans Shinken afin de prendre en compte l'activation de l'authentification.

## Étape 1 : Créer l'utilisateur d'authentification à la base

On commence par créer l'utilisateur dans MongoDB qui sera utilisé pour s'authentifier.

Se connecter au shell MongoDB :

```
mongo
```

Depuis le shell MongoDB, exécutez les deux commandes suivantes :

```
use admin
```

```
db.createUser(
 { user : 'YOUR_USER',
   pwd : 'YOUR_PASSWORD',
   roles : [ { role : 'root', db : 'admin' } ]
 }
)
```

Adapter le nom d'utilisateur ( *YOUR_USER* ) et le mot de passe ( *YOUR_PASSWORD* ) dans la commande.

> Il ne faut pas modifier les champs **role** et **db.**
>
> Pour que Shinken fonctionne correctement, l'utilisateur doit disposer de privilèges avancés sur l'ensemble des bases de données, ce qui impose la valeur de ces deux paramètres.

Exemple :

```
mongos> db.createUser(
...  { user : 'shinken',
...    pwd : 'shinken',
...    roles : [ { role : 'root', db : 'admin' } ]
...  }
... )
Successfully added user: {
        "user" : "shinken",
        "roles" : [
                {
                        "role" : "root",
                        "db" : "admin"
                }
        ]
}
```

## Se connecter à MongoDB avec l'utilisateur crée

Il est désormais possible de s'authentifier lors de la connexion à MongoDB.

Les identifiants peuvent être spécifiés directement dans la commande de lancement du shell MongoDB :

```
mongo --username YOUR_USER --password YOUR_PASSWORD --authenticationDatabase admin
```

> Si le champ --password est laissé vide, un prompt s'affiche pour demander le mot de passe, évitant ainsi d'exposer celui-ci en clair.

Il est également possible de fournir les identifiants une fois connecté dans le shell MongoDB :

```
use admin
db.auth('YOUR_USER', 'YOUR_PASSWORD')
```

> La commande db.auth n'est pas sauvegardée dans l'historique des commandes du shell MongoDB, ce qui évite d'exposer le mot de passe.

Lorsque l'authentification par mot de passe sera activée dans MongoDB, seules les connexions avec des identifiants valides seront acceptées.

## Changer le mot de passe d'un utilisateur

Pour changer le mot de passe de l'utilisateur créé , il faut se connecter au shell MongoDB avec les identifiants actuels , puis exécuter les deux commandes suivantes :

```
use admin
```

```
db.changeUserPassword('YOUR_USER', 'NEW_PASSWORD')
```

## Étape 2 : Activer l'authentification à la base

Pour activer l'authentification dans MongoDB, il faut ajouter le champ suivant dans le fichier de configuration /etc/mongod.conf

```
security:
  authorization: enabled
```

### Exemple de fichier de configuration complet

#### RHEL / CentOS 7 ou RHEL / Alma / Rocky 8 ou RHEL / Alma / Rocky 9

```
# for documentation of all options, see:
#   http://docs.mongodb.org/manual/reference/configuration-options/

# where to write logging data.
systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongodb/mongod.log

# Where and how to store data.
storage:
  dbPath: /var/lib/mongo
  journal:
    enabled: true

# how the process runs
processManagement:
  fork: true  # fork and run in background
  pidFilePath: /var/run/mongodb/mongod.pid

# network interfaces
# NOTE: when go as a replicat member (cluster), change the 27017 to 27018 according to configuration
#       and comment the bindIp parameter
net:
  port: 27017
  unixDomainSocket:
    enabled: false
  bindIp: 127.0.0.1  # Listen to local interface only, comment to listen on all interfaces.

storage:
    engine: wiredTiger

security:
 authorization: enabled
```

#### Debian 13

Sur une distribution **Debian** il faut noter que :

- la base de données est stockée dans le dossier **/var/lib/mongodb**.
- Le système de démarrage des démons est géré différemment, il ne doit **pas** y avoir **de section processManagement** dans le fichier de configuration.

```
# for documentation of all options, see:
#   http://docs.mongodb.org/manual/reference/configuration-options/

# where to write logging data.
systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongodb/mongod.log

# Where and how to store data.
storage:
  dbPath: /var/lib/mongodb
  journal:
    enabled: true

# network interfaces
# NOTE: when go as a replicat member (cluster), change the 27017 to 27018 according to configuration
#       and comment the bindIp parameter
net:
  port: 27017
  unixDomainSocket:
    enabled: false
  bindIp: 127.0.0.1  # Listen to local interface only, comment to listen on all interfaces.

storage:
    engine: wiredTiger

security:
 authorization: enabled
```

# Étape 3 : Déclarer l'utilisateur et le mot de passe dans Shinken

## Dans les fichiers de configuration

Il est désormais possible de déclarer l'utilisateur et le mot de passe à utiliser pour que Shinken s'authentifie à la base.

- Tous les composants de Shinken qui se connectent à MongoDB doivent voir leur configuration modifiée sur le serveur de l'Arbiter.
- Voici la liste des éléments qui se connectent à MongoDB et dont la configuration doit être mise à jour :
  - Le démon Synchronizer : ( *voir la page* Paramètres globaux ( synchronizer.cfg ) ) ;
  - Le module event-manager-reader : ( *voir la page* Module event-manager-reader ) ;
  - Le module event-manager-writer : ( *voir la page* *Module event-manager-writer* ) ;
  - Le module Graphite-Perfdata : ( *voir la page* Module Graphite-Perfdata ) ;
  - Le module MongoDB : ( *voir la page* Module MongoDB ) ;
  - Le module MongodbRetention : ( *voir la page* Module MongodbRetention ( Rétention en base de données centralisée par royaume ) ) ;
  - Le module SLA du Broker : ( *voir la page* Module SLA ) ;
  - Le module SLA de la WebUI : ( *voir la page* Module SLA ( WebUI ) ) ;
  - Le module livedata-module-sla-provider ( *voir la page* Module livedata-module-sla-provider ) ;
  - Le collecteur de type discovery-import ( *voir la page* Collecteur de type discovery-import ( Scan NMAP ) ) ;
  - Le module report-builder--module-sla-reader ( *voir la page* Module report-builder--module-sla-reader ) ;
- Dans le cas de l'utilisation de l'outil tiers Grafana, il faut aussi modifier sur la ou les machines avec un carbon-cache le fichier de configuration **/opt/graphite/conf/mongodb.conf** ( *voir la page* Grafana - v8.3.2 ) ;

## Dans les commandes

Une fois que l'authentification par mot de passe est activée, il faut de fournir les identifiants aux commandes Shinken qui se connectent à la base. L'aide des commandes permet de savoir si une commande se connecte à MongoDB.

Les paramètres pour s'identifier sont : **--mongo-username** et **--mongo-password**

Avant d'exécuter tout traitement, les commandes Shinken se connectant à MongoDB vérifient d'abord qu'elles disposent des privilèges nécessaires, garantissant ainsi l'intégrité et la cohérence de Shinken

Si l'option **--mongo-password** est utilisée, le mot de passe risque d'être visible dans l'historique des commandes ( *via la commande **history** par exemple* ).

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 faire un fichier qui contient le mot de passe et d'utiliser ce fichier dans la commande ( *en utilisant **cat** ou les directives de redirections des entrées sorties du shell par exemple* ) : --mongo-password $(<my_file).

## Étape 4 : Redémarrer MongoDB et Shinken

Pour appliquer l'activation de l'authentification, il faut redémarrer mongod et Shinken

Il faut éteindre dans l'ordre :

- Shinken

```
service-shinken stop
```

- MongoDB

```
service mongod stop
```

Ensuite, il faut redémarrer dans l'ordre suivant :

- MongoDB

```
service mongod start
```

- puis Shinken

```
service-shinken start
```

L'authentification est désormais activée. MongoDB autorisera uniquement les connexions de l'utilisateur créé.

Si Grafana est utilisé pour afficher les métriques, il faut aussi redémarrer apache sur les serveurs avec un carbon-cache :

### RHEL / CentOS 7 ou RHEL / Alma / Rocky 8 ou RHEL / Alma / Rocky 9

```
systemctl restart httpd
```

### Debian 13

```
systemctl restart apache2
```

## Étape 5 : Mise à jour de la supdesup de Shinken

L'activation de l'authentification par mot de passe sur MongoDB n'est pas automatiquement prise en compte par la supervision.

- Les checks **Broker - DB - Last Flush Time**, **Broker - DB - Open Connections**, **Synchronizer - DB - Last Flush Time**, **Synchronizer - DB - Open Connections** seront en erreurs.
- Il est nécessaire de modifier les modèles d'hôtes **shinken-synchronizer** et **shinken-broker** pour y renseigner les identifiants. Les données à remplir sont :
  - DB_AUTH_DB
  - DB_USER_NAME

- DB_USER_PASSWORD