Activer l'authentification par mot de passe dans un cluster MongoDB

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** s e connectant à MongoDB.
- Ces identifiants devront être **aussi** fournis lors de **l'utilisation de toute commande Shinken** nécessitant un **accès** à la base de données.

Cette procédure suppose de partir d'un Cluster MongoDB (*voir la page* *Haute disponibilité de la base MongoDB (mise en place d'un cluster)*) sans qu'aucun utilisateur n'ait été encore crée.

- Pour sécuriser les communications entre les nœuds du cluster, il est recommandé d'activer le chiffrement SSL (*voir la page* Activer le chiffrement (SSL) pour les communications d'un cluster MongoDB).
  - Le chiffrement peut être activé avant ou après l'activation de l'authentification.

- La procédure décrite permet de minimiser au maximum le temps d'interruption de service.
  - Le 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 le cluster MongoDB.
3. Configurer Shinken pour s'authentifier avec l'utilisateur auprès de MongoDB.
4. Redémarrer le cluster MongoDB et Shinken pour activer l'authentification.
5. Mettre à jour la supervision 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 Mongo qui sera utilisé pour s'authentifier.

L'utilisateur doit être créé deux fois :

- une fois pour tous les **mongos**, permettant de manipuler les données,
- une fois pour tous les **mongod**, pour les opérations de maintenance du cluster.

Créer l'utilisateur pour les mongos

Cette étape est à réaliser une unique fois sur un seul des mongos. L'utilisateur crée sera automatiquement diffusé sur les autres mongos et les mongo-configsrv.

Se connecter au shell mongo d'un des mongos du cluster.

```
mongo --port 27017
```

Adapter le port de la commande au port du mongos (*si nécessaire*).

Depuis le shell MongoDB, exécuter 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"
                }
        ]
}
```

Créer l'utilisateur pour les mongod

Cette étape est à réaliser une unique fois sur le **mongod primaire**. L'utilisateur crée sera automatiquement diffusé sur les autres mongod du cluster.

Se connecter au shell mongo du **mongod primaire**.

```
mongo --port 27018
```

Adapter le port de la commande au port du mongod (*si nécessaire*).

Depuis le shell MongoDB, exécuter 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. **U tiliser les mêmes identifiants que lors de la création de l'utilisateur dans le mongos.**

Il ne faut pas modifier les champs **role** et **db.**

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 Mongo :

```
 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 Mongo :

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

La commande db.auth n'est pas sauvegardée dans l'historique des commandes du shell Mongo, 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 mongo avec les identifiants actuels , puis exécuter les deux commande s suivantes :

```
use admin
```

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

Étape 2 : Activer l'authentification à la base

Génération de la clé d'authentification interne

Lancer les commandes suivantes pour générer la clé partagée :

```
mkdir /srv/mongodb/
/opt/shinken/openssl/bin/openssl rand -base64 741 > /srv/mongodb/mongodb-keyfile
```

Les commandes suivantes permettent d'accorder les bons droits au fichier :

```
chmod 600 /srv/mongodb/mongodb-keyfile
chown mongod:mongod /srv/mongodb/mongodb-keyfile
```

Les commandes suivantes permettent d'accorder le bon contexte **selinux** au fichier :

```
semanage fcontext -a -t mongod_var_lib_t /srv/mongodb/mongodb-keyfile
restorecon -v /srv/mongodb/mongodb-keyfile
```

Il faut ensuite déployer la clé sur **tous les serveurs avec un mongos, mongod, ou mongo-configsrv :**

```
rsync -avPAX /srv/mongodb/ node:/srv/mongodb/
```

Remplacer **node** par les noms des différents serveurs où tournent les processus mongod, mongos ou mongo-configsrv.

Activer l'authentification du cluster

Lorsque l'authentification est activée sur un cluster, chaque nœud doit fournir des identifiants pour s'authentifier auprès des autres nœuds, ce qui correspond à l'authentification interne.

- L'activation de l'authentification interne permet également de mettre en place un contrôle d'accès par mot de passe.
- Une fois cette authentification activée, seul l'utilisateur créé pourra se connecter.

Pour activer l'authentification dans un cluster, il est nécessaire d'activer l'authentification individuellement sur chaque nœud :

- Cela inclut l'activation de l'authentification sur chaque **mongos, mongod, mongo-configsvr** qui composent le cluster.
- L'authentification interne repose sur une clé secrète partagée entre les différents nœuds du cluster.

Ensuite, pour **tous** les processus mongod, mongos et mongo-configsrv, ajoutez le champ suivant dans leurs fichiers de configuration respectifs (*/etc /mongo-configsrv.conf, /etc/mongod.conf, /etc/mongos.conf*)

```
security:
  keyFile: /srv/mongodb/mongodb-keyfile
```

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 rediriger le contenu d'un fichier contenant le mot de passe (*en utilisant* ***cat*** *par exemple*) : --mongo-password $(cat my_file).

Étape 4 : Redémarrer le cluster MongoDB et Shinken

Pour appliquer l'activation de l'authentification, il faut redémarrer les démons de mongo et Shinken **sur tous les serveurs** .

Il faut éteindre dans l'ordre :

- d'abord les démons Shinken :

```
service-shinken stop
```

- puis les démons mongo :

```
service mongod stop
```

```
service mongo-configsrv stop
```

```
service mongos stop
```

Ensuite, il faut redémarrer dans l'ordre suivant :

- les démons mongo :

```
service mongod start
```

```
service mongo-configsrv start
```

```
service mongos start
```

- ensuite les démons Shinken :

```
service-shinken start
```

L'authentification est désormais activée.

- Mongo autorisera uniquement les connexions de l'utilisateur créé pour Shinken.

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

```
service httpd restart
```

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_USER_NAME
        - DB_USER_PASSWORD