


Gestion d'un cluster avec des éléments venant de royaumes différents

Sommaire

- [Contexte](#)
- [Architecture](#)
- [Installation - Les étapes de mise en place](#)
 - [Mise en place de l'architecture Shinken sur le réseau isolé](#)
 - [Paramétrage sur le réseau Central](#)
- [Troubleshoot](#)
 - [Commande manuelle](#)
 - [Réseau](#)

Description

Le module event-manager-writer est un module de la fonctionnalité [bac à événements](#) qui permet l'écriture des événements en base de donnée .

 Pour que la fonctionnalité bac à événement fonctionne il faut absolument que ce module soit activé.

  Il ne peut y avoir qu'un event-manager-writer par base Mongo. Donc par exemple avec 2 broker sur la même machine soit vous n'activez le module que sur un broker soit vous configurez le module pour écrire dans une autre base.

Configuration du fichier cfg

Voici le fichier CFG de configuration présent dans : `/etc/shinken/modules/event_manager_writer.cfg`

```
#####
# event manager
#####
# Daemons that can load this module:
# - broker (to save events information into a mongodb database)
# This module compute and save event for event manager
#####

define module {

    # #
    #     MODULE IDENTITY     #
    # #

    # Module name [ Must be unique ]                               [ MANDATORY ]
    module_name                                     event-manager-writer

    # Module type [ Do not edit ]                                   [ MANDATORY ]
    module_type                                     event_container

    # #
    #     MODULE OPTIONS     #
    # #

    # Number of day the events are keep in database
    # >>> Default : 30 ( days )
    # day_keep_data                                     30

    # #
    #     DATABASE CONNECTION     #
    # #

    # MongoDB parameters
    # MongoDB uri definition . You can find the mongodb uri syntax at
```

```

# https://docs.mongodb.com/manual/reference/connection-string/
# >>> Default : mongodb://localhost/?w=1&fsync=false
# uri                                mongodb://localhost/?w=1&fsync=false

# Which database contains events data
# >>> Default : event_container
# database                            event_container

# SSH tunnel activation to secure your mongodb connection
# That will allow all mongodb to be encrypted & authenticated with SSH
# Enable : 1 ( enable ssh tunnel )
# >>> Disable : 0 ( disable ssh tunnel ) ( Default )
# use_ssh_tunnel                      0

# If the SSH connection goes wrong, then retry use_ssh_retry_failure time before_shinken_inactive
# >>> Default : 1 ( try )
# use_ssh_retry_failure               1

# SSH user to connect to the mongodb server.
# >>> Default : shinken
# ssh_user                            shinken

# SSH keyfile to connect to the mongodb server.
# >>> Default : ~shinken/.ssh/id_rsa

# SSH Timeout used to test if the SSH tunnel is viable or not, in seconds.
# >>> Default : 10 ( seconds )
# ssh_tunnel_timeout                 10

# AutoReconnect Management
# When MongoDB require you to reconnect ( For example, It can occur when a new PRIMARY is elected
# in a MongoDB cluster ), it will raised the MongoDB AutoReconnect exception.
#
# How many try to reconnect before module go in error
# >>> Default : 4 ( try )
# auto_reconnect_max_try              4

# Time between each try
# >>> Default : 3 ( seconds )
# auto_reconnect_sleep_between_try    3

# NOTE: Change these values only if you have a MongoDB cluster and you change the
# heartbeatTimeoutSecs of your MongoDB replica set
# The value of auto_reconnect_max_try * auto_reconnect_sleep_between_try must be higher than
# heartbeatTimeoutSecs in the rs.conf(); of your MongoDB replica set.

# #
# WORKERS IN THE BROKER #
# #

# This module will use workers in the broker, each worker will manage a shard of all hosts/checks.
# This parameter is used by the broker to set the number of workers.
# Each worker will use one CPU, which will balance the event processing load among CPUs.
# >>> Default : 1 ( worker )
# broker_module_nb_workers           1

# #
# INTERNAL OPTIONS #
# #

# INTERNAL : DO NOT EDIT FOLLOWING PARAMETER WITHOUT YOUR DEDICATED SUPPORT

# Broker idle time before considering that Shinken is inactive.
# Use this if you have Broker loop time that exceeds 30 seconds
# >>> Default : 30 ( seconds )
# time_before_shinken_inactive       30
}

```

Option du module

Taille du bac d'événement en nombre de jours

Le paramètre "day_keep_data" permet de choisir le nombre de jours qu'un événement est gardé dans votre base.

Si votre base MongoDB prend trop de place sur le disque, (à monitorer avec le check : [shinken-broker-module-event-manager-writer](#)) . Il est possible de diminué le nombre de jours sauvegardé.

| Nom du paramètre | Valeur par défaut | Description |
|------------------|-------------------|---|
| day_keep_data | 30 | Durée en nombre de jour d'un événement dans le bac à événement. |

Accès à la base MongoDB

Cette configuration s'effectue dans le fichier de configuration du module.

Pour se connecter à la base MongoDB utilisé pour le stockage des données, 2 méthodes sont disponibles:

- **Connexion directe:** Par défaut, mais non sécurisée.
- **Tunnel SSH:** Shinken se connecte à la base MongoDB au travers d'un module SSH pour plus de sécurité

Connexion directe au serveur Mongo

Par défaut, le module se connecte de manière directe à la base MongoDB pour y lire et écrire les données.

Dans la configuration du module, on sait que la connexion se fait de manière directe lorsque le paramètre "use_ssh_tunnel" est à 0.

Cette méthode de connexion a pour avantage d'être facile à configurer au niveau de Shinken. Par contre, elle oblige à permettre l'accès à la base MongoDB au monde extérieur, et donc s'exposer à des problèmes de sécurité.

La sécurisation de la base MongoDB est bien sur toujours possible (voir [Sécurisation des connexions aux bases MongoDB](#)) mais bien plus complexe à mettre en place. La méthode de connexion par SSH est donc préférable pour des raisons pratiques et de sécurité.

Connexion par SSH au serveur Mongo

Le module peut également se connecter par tunnel SSH à la base MongoDB, pour des raisons de sécurité.

En effet, le paramétrage de MongoDB permet de définir sur quel interface réseau ce dernier écoute les requêtes. En n'autorisant seulement interface réseau avec l'adresse 127.0.0.1, cela évite d'ouvrir la base au monde extérieur.

Dans la configuration de la base MongoDB (/etc/mongod.conf), assurez-vous que le paramètre "bind_ip" est positionné pour n'écouter que sur l'interface locale:

- bind_ip=127.0.0.1

Dans cette configuration la base MongoDB écoute que sur l'interface réseau local, pour que le module se connecte, il faut passer par un tunnel SSH. Pour ce faire activer les options suivantes :

| Nom du paramètre | Valeur par défaut | Description |
|-----------------------|----------------------|--|
| use_ssh_tunnel | 0 | <ul style="list-style-type: none">• 1 : Connection par tunnel SSH• 0 : Connection direct |
| use_ssh_retry_failure | 1 | Spécifie le nombre supplémentaire de tentatives lors de l'établissement du tunnel SSH si ce dernier n'arrive pas à être établi |
| ssh_user | shinken | L'utilisateur avec lequel le tunnel sera établi |
| ssh_keyfile | ~shinken/.ssh/id_rsa | La clé ssh privée présent sur le serveur Shinken qui sera utilisé pour établir le tunnel. |
| ssh_tunnel_timeout | 10 | Spécifie le timeout en secondes de la vérification du tunnel SSH avant que la connexion vers MongoDB soit effectuée |

Le tunnel SSH va permettre au module de se connecter comme si ses requêtes étaient local à la base MongoDB (en 127.0.0.1)

- Connectez-vous avec le user lançant le démon sur le serveur Shinken
- Générez la paire de clés SSH si nécessaire
- Copiez la clé publique sur le serveur mongo

```
root@serveur_shinken # su - shinken
shinken@serveur_shinken $ ssh-keygen
shinken@serveur_shinken $ ssh-copy-id user_distant@serveur_mongo
[...]
shinken@serveur_shinken $ ssh user_distant@serveur_mongo
user_distant@serveur_mongo $
```

- Cette manipulation est aussi nécessaire dans le cas ou la base MongoDB est sur le même serveur que le module, même si le tunnel est ouvert localement.

Gestion de l'auto reconnexion avec un cluster MongoDB



Définitions

- **Primaire**: nom de Mongo pour désigner un serveur maître, le serveur sur lequel il est possible de faire des requête d'écriture dans la base.
- **Election** : processus de Mongo pour choisir un nouveau membre Primaire si le membre Primaire devient inaccessible

Voir : [Haute disponibilité de la base Mongo](#)

Dans le cas de l'utilisation d'un cluster MongoDB, lorsque le membre Primaire devient inaccessible une nouvelle élection est déclenché ce qui provoque une coupure temporaire de l'accès à la base.

Dans le but de ne pas interrompre le service, le module SLA va se reconnecter automatiquement au cluster MongoDB. Pour ce faire il va faire un nombre d'essais égale au paramètre "auto_reconnect_max_try" avec une pause de X secondes entre chaque essais (correspondant au paramètre "auto_reconnect_sleep_between_try").

Par défaut pour MongoDB le temps maximum avant qu'un membre Primaire soit considéré comme indisponible et qu'une nouvelle élection ait lieu est de 10 secondes.

Voir : "heartbeatTimeoutSecs" donné par la commande rs.conf(); dans un shell de mongo.

| Nom du paramètre | Valeur par défaut | Description |
|----------------------------------|-------------------|--|
| auto_reconnect_max_try | 4 | Nombre d'essais de reconnexion à la base |
| auto_reconnect_sleep_between_try | 3 | Temps entre chaque essais en seconde |

Les valeurs par défauts du fichier laisse 12 secondes, ce qui est amplement suffisant avec la configuration par défaut de MongoDB.



Il est conseillé de ne pas modifier ces valeurs.

Utilisation des workers du module event-manager-writer

Le paramètre "broker_module_nb_workers" va déterminer combien de fois le module va se cloner pour gérer le flux de donnée à enregistrer afin de repartir cette charge sur plusieurs CPU.

Il est possible de changer ce paramètre si l'utilisation CPU du processus : "*NOM DU BROKER* [- Module: event-manager-writer][Worker: 0]" est trop élever.

| Nom du paramètre | Valeur par défaut | Description |
|--------------------------|-------------------|---|
| broker_module_nb_workers | 1 | Nombre de workers (nombre de clone du module) qui traite le flux de donnée pour sauvegarder les événements dans la base MongoDB |



Ne pas dépassé le nombre de core cpu de la machine : cela serais contre productif pour les performances.