

shinken-synchronizer (Modèle d'hôte)

Sommaire

[Contexte](#)
[Description du modèle](#)
 [Checks](#)
 [Paramètre du check](#)
 [Détail des commandes](#)
[Description des erreurs de Synchronizer - \\$KEY\\$ - Alive](#)
 [Erreur de surcharge des disques de logs](#)
 [Erreur de vol de CPU](#)
 [Erreur d'un démon bloqué, qui doit être redémarré](#)
 [Le démon a bloqué une tentative de chargement d'objet malveillant](#)
 [Le démon est en cours d'arrêt](#)

Description

Le module `event-manager-reader` est un module qui permet au module WebUI (l'interface de visualisation) du broker d'afficher le [bac à événements](#).

- Il récupère les événements en base de donnée des événements.
- Il doit donc être listé dans la liste des modules du module WebUI (cf [Module WebUI](#), propriété "modules")



Pour que la fonctionnalité bac à événement apparaisse dans l'interface, il faut absolument que ce module soit activé.



Si vous avez plusieurs configurations de WebUI présente dans votre architecture, n'oubliez pas d'activer le module dans chaque configuration.

Exemple :

- Plusieurs WebUI sur le même Broker.
- Une WebUI sur différents Brokers qui n'utilise pas la même définition.

Activation du module

Le module `event-manager-reader` est un module qui peut être activé seulement sur le module WebUI.

- L'activation du module s'effectue en ajoutant le nom de ce module dans le fichier de configuration du module WebUI.
- Pour ce faire, ouvrir le fichier de configuration du Broker à l'emplacement `/etc/shinken/brokers/nom_du_broker.cfg`, et ajouter le nom de votre module "WebUI".
- Une fois le module WebUI ajouté au Broker, ajouter votre module "event-manager-reader" au module WebUI à l'emplacement `/etc/shinken/modules/ma_webui.cfg`.

Exemple: par défaut, nous livrons un module dont le nom est " `event-manager-reader`":

```
define module {
    [...]
    modules          Module 1, Module 2, Module 3, event-manager-reader
    [...]
}
```

Pour prendre en compte le changement de configuration, redémarrer l'Arbiter:

```
service shinken-arbiter restart
```

Configuration

La configuration du module que nous livrons par défaut se trouve dans le fichier `/etc/shinken/modules/event_manager_reader.cfg`

- Vous trouverez aussi systématiquement un exemple dans `/etc/shinken-user-example/configuration/daemons/brokers/modules/event_manager_reader/event_manager_reader.cfg`

Exemple de fichier de configuration

`/etc/shinken/modules/event-manager-reader.cfg`

```
#####
# event manager
#####
# Modules that can load this module:
# - WebUI (to display events data to the users)
# This module read event from database to show you in the WebUI
#####

define module {

    # #
    #     MODULE IDENTITY     #
    # #

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

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

    # #
    #     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 )
    #     Default : Disable => 0 ( disable ssh tunnel )
    #
    # 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_keyfile                                     ~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.

# #
#   VISUALISATION UI           #
# #

# Specify the max retries when the Visualisation UI can't communicate with the event_manager_reader
# during the Excel export.
# Increase this parameter when users encounter difficulties when generating the Excel export
# of events. This will make the export more tolerant to network loss or database slowness,
# or restart of the broker.
# The value cannot be 0.
#
#         Default : 2 ( try )
#
# broker__module_webui__module_event_manager_reader__events_export__max_retry 2

# #
#   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
}

```

Détails des sections composant le fichier de configuration

Identification du module

Il est possible de définir plusieurs instances de module de type "event-manager-writer" dans votre architecture Shinken.

- Chaque instance devra avoir un nom unique.

Nom	Type	Unité	Défaut	Commentaire
module_name	Texte	---	event-manager-reader	Nous vous conseillons de choisir un nom en fonction de l'utilisation du module pour que votre configuration soit simple à maintenir. Doit être unique.
module_type	Texte	---	event_container	Ne peut être modifié.

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 MongoDB

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 sûr 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.

```
# 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
```

Nom	Type	Unité	Défaut	Commentaire
uri	Texte	URL	mongodb://localhost/?w=1&fsync=false	Vous pouvez trouver la syntaxe de l'uri de MongoDB à l'adresse https://docs.mongodb.com/manual/reference/connection-string/
database	Texte	---	event_container	Nom de la base de données où sont stockés les données événements

Connexion par SSH au serveur MongoDB

```

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

```

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 quelle 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	Type	Unité	Défaut	Commentaire
<code>use_ssh_tunnel</code>	Booléen	---	0	<ul style="list-style-type: none"> • 1 : Connexion par tunnel SSH • 0 : Connexion directe
<code>use_ssh_retry_failure</code>	Entier	Nombre d'essais	1	Spécifie le nombre supplémentaire de tentatives lors de l'établissement du tunnel SSH si ce dernier n'arrive pas à être établi
<code>ssh_user</code>	Texte	Utilisateur unix	shinken	L'utilisateur avec lequel le tunnel sera établi

ssh_keyfile	Texte	Chemin de fichier	~shinken/.ssh/id_rsa	La clé ssh privée présente sur le serveur Shinken qui sera utilisé pour établir le tunnel.
ssh_tunnel_timeout	Entier	Ssecondes	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 l'utilisateur lançant le démon sur le serveur Shinken
- Générez la paire de clés SSH si nécessaire
- Copiez la clé publique sur le serveur MongoDB

- **Copie de la clé SSH**

```
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 où 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

```
# 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.
```

Définitions

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

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

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 égaux au paramètre "auto_reconnect_max_try" avec une pause de X secondes entre chaque essai (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 MongoDB.

Nom	Type	Unité	Défaut	Commentaire
<code>auto_reconnect_max_try</code>	Entier	Nombre d'essais	4	Nombre d'essais de reconnexion à la base
<code>auto_reconnect_sleep_between_try</code>	Entier	Secondes	3	Temps entre chaque essai 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.

Configuration pour l'interface de visualisation

```
# #
# VISUALISATION UI #
# #

# Specify the max retries when the Visualisation UI can't communicate with the event_manager_reader
# during the Excel export.
# Increase this parameter when users encounter difficulties when generating the Excel export
# of events. This will make the export more tolerant to network loss or database slowness,
# or restart of the broker.
# The value cannot be 0.
#
# Default : 2 ( try )
#
# broker__module_webui__module_event_manager_reader__events_export__max_retry 2
```

Le paramètre "broker__module_webui__module_event_manager_reader__events_export__max_retry" permet de configurer le nombre maximal de tentatives lorsque l'interface de visualisation ne peut pas communiquer avec le module event_manager_reader pendant l'export des données vers Excel.

Cela rendra les exports plus tolérants aux pertes de réseau ou aux lenteurs de la base de données, ou aux redémarrages du Broker.

Nom	Type	Unité	Défaut	Commentaire
<code>broker__module_webui__module_event_manager_reader__events_export__max_retry</code>	Entier	Nombre d'essais	2	Nombre d'essais d'appel pour l'export des données vers Excel. Ne peut pas valoir 0.

Options Internes

```

# #
#   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

```



Ces paramètres sont dédiés au fonctionnement interne au module, il est fortement recommandé de ne pas les modifier sans votre support dédié.

Nom	Type	Unité	Défaut	Commentaire
time_before_shinken_inactive	Entier	Secondes	30	Temps d'inactivité du Broker avant de considérer que Shinken est inactif. Utilisez cette option si vous avez un temps de boucle du Broker qui dépasse 30 secondes.