

Module MongoDB

Explication

Le module MongoDB permet de configurer l'accès à la base MongoDB par WebUI.

Configuration

Voici le fichier CFG de configuration présent dans : **/etc/shinken/modules/mongodb.cfg**

```
#####
# Mongodb
#####
# Modules that can load this module:
# - WebUI
# This module is mandatory for the WebUI module. The WebUI use this definition to access the
# mongodb data (like hive and user shares)
#####

define module {
# Shinken Enterprise. Lines added by import core. Do not remove it, it's used by Shinken Enterprise to
update your objects if you re-import them.
    _SE_UUID          core-module-5dfa7f645adc11e5ad8e080027f08538
    _SE_UUID_HASH     9068b8f577e092efe14d185538fbdb6d
# End of Shinken Enterprise part

    ##### Module identity #####
    # Module name. Must be unique
    module_name      Mongodb

    # Module type (to load module code). Do not edit.
    module_type      mongodb

    ##### Database connection #####
    # mongodb uri definition for connecting to the mongodb database. You can find the mongodb uri
    # syntax at https://docs.mongodb.com/manual/reference/connection-string/
    uri              mongodb://localhost/?safe=true

    # If you want to securize your mongodb connection you can enable the ssh use_ssh_tunnel that will
    # allow all mongodb to be encrypted & authenticated with SSH
    # Should use a SSH tunnel (Default 0=False)
    # use_ssh_tunnel      0

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

    # SSH user/keyfile in order to connect to the mongodb server.
    # Default: shinken
    # ssh_user            shinken

    # Default: ~shinken/.ssh/id_rsa
    # ssh_keyfile         ~shinken/.ssh/id_rsa

    # Timeout in order to establish a connection, in seconds
    # Default: 10
    # mongo_timeout      10

    # Which database is used to store sla data
    database            shinken
}

```

Configurer l'accès à la base MongoDB

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



Les données de la WebUI sont stockées dans la base Mongo locale au Broker

Pour se connecter au serveur Mongo utilisé pour le stockage des données de la WebUI, 2 méthodes sont disponibles:

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

Connexion directe au serveur Mongo

Par défaut, le module MongoDB se connecte de manière directe au serveur Mongo pour y lire et écrire les données de la WebUI.

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

```
define module {
...
...

#==== Module identity ====
# Module name. Must be unique
module_name    Mongodb
...
use_ssh_tunnel 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 Mongo au monde extérieur, et donc s'exposer à des problèmes de sécurité.

La sécurisation de la base Mongo est bien sur toujours possible (voir la page [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 MongoDB peut également se connecter par tunnel SSH au serveur Mongo, pour des raisons de sécurité.

- En effet, le paramétrage de mongoDB permet de définir sur quel adresse ce dernier écoute les requêtes.
 - En n'autorisant seulement l'adresse 127.0.0.1, cela évite d'ouvrir la base au monde extérieur.
 - Dans la configuration du serveur Mongo (/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, pour que le module mongoDB se connecte, il faut activer les options suivantes :

Nom du paramètre	Description
mongodb_use_ssh_tunnel	Activation du tunnel ou non (0 ou 1). Par défaut, il n'est pas activé (à 1)
mongodb_use_ssh_retry_failure	Spécifie le nombre supplémentaire de tentatives lors de l'établissement du tunnel SSH si ce dernier n'arrive pas à être établi
mongodb_ssh_user	L'utilisateur avec lequel le tunnel sera établi
mongodb_ssh_keyfile	une clé ssh privée présent sur le serveur Shinken (par défaut ~/.ssh/id_rsa) qui sera utilisé pour établir le tunnel.

- Le tunnel SSH va permettre au module de se connecter comme si ses requêtes étaient local au serveur mongo (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 où la base mongoDB est sur le même serveur que le module SLA, même si le tunnel est ouvert localement.