

Module SLA

Sommaire

Contexte	Architecture du module d'écriture
Activation du module	Exemple d'activation du module nommé "sla" sur le démon "broker-master" (configuration livrée par défaut par Shinken)
	Créer un nouveau module de type SLA
Configuration	Exemple de fichier de configuration
	Détails des sections composant le fichier de configuration
	Identification du module
	Suppression des anciennes entrées dans la base d'archives du module SLA
	Sauvegarde des SLA brut
	Option de calcul du taux de SLA
	Option de stockage des Resultats et Resultats longs
	Accès à la base MongoDB
	Configuration de l'URI de connexion et de l'authentification par mot de passe
	Connexion directe au serveur MongoDB
	Connexion par SSH au serveur MongoDB
	Gestion de l'auto reconnexion avec un cluster MongoDB
	Utilisation des workers du module SLA
	Options internes
Erreurs dans le Module	

Contexte

Le module SLA peut être accroché à deux endroits : le Broker et la WebUI.

L'accrocher sur le Broker permet :

- De calculer les valeurs de SLA (*Service Level Agreement*) des éléments supervisés et les stocker dans la base de données Mongodb définie dans le fichier de configuration ci-dessous.
- De modifier la méthode de calcul des SLA (*par exemple, choisir de considérer un statut "Avertissement" comme une période positive de SLA, ou encore d'exclure les périodes de maintenance dans le calcul*).



Les deux modules étant complémentaires (le module SLA sur le Broker étant l'écrivain et celui de la WebUI étant le lecteur), Shinken fournit **un seul** fichier de configuration commun entre les deux, ceci permettant de garder une cohérence entre l'écriture et la lecture des SLA.

Avoir le même fichier, évitera de répéter la même valeur dans deux fichiers de configuration dans le cas des paramètres communs.

Architecture du module d'écriture

Le module SLA d'écriture se compose de trois parties, incarné par trois processus :

- Worker(s)
 - Ce processus se charge de gérer le flux de données, qu'il enregistre sous un format brut (*raw_sla*).
 - Il est possible d'ajouter des Workers pour gérer le flux de données.
- Archive
 - Ce processus se charge d'agréger les données au format brut (*raw_sla*) pour accélérer la lecture et réduire l'espace de stockage nécessaire.
- Migration
 - Ce processus se charge de mettre à jour le format des entrées à la suite d'une mise à jour.
 - Il s'occupe aussi de la rotation des données (*voir le chapitre : Sauvegarde des SLA brut*).

Activation du module

Les modules de type "s la" sont des modules qui doivent être activés sur le démon Broker, qu'on appellera le **démon parent**.

- L'activation du module s'effectue en ajoutant le **nom** du module dans la configuration du **démon parent**.
 - Pour cela, il faut ouvrir le fichier de configuration du **démon parent** et ajouter dans le paramètre **démon**, le nom du module de type "s la".
- Il est possible de faire plusieurs modules de type "s la".

- Cela permet, **par exemple**, d'avoir des configurations différentes en fonction des royaumes.
- **Contraintes :**
 - Ce module est activable sur un démon Broker, mais aussi sur les modules de type webui (voir la page [Module WebUI](#)).
 - Sur une architecture distribuée, il faut qu'un seul des démons de type Broker ait un module de type par "s la" par royaume.
 - Il est possible d'activer plusieurs modules de type "s la" sur un Broker pour les faire écrire dans différentes bases (*réplication*), mais **Shinken conseille FORTEMENT** d'utiliser la réplication par cluster Mongo à la place (voir la page [Haute disponibilité de la base MongoDB \(mise en place d'un cluster\)](#)).

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

```
service-shinken-arbiter restart
```

Exemple d'activation du module nommé "sla" sur le démon "broker-master" (configuration livrée par défaut par Shinken)

L'exemple suivant

- active le module "sla",
- sur le démon "broker-master", dont la configuration est dans le fichier `/etc/shinken/brokers/broker-master.cfg`.

Modification dans le fichier du module `/etc/shinken/brokers/broker-master.cfg` :

```
define broker {
    [...]
    modules          Module 1, Module 2, Module 3, sla
    [...]
}
```

Puis redémarrage de l'Arbiter

```
service-shinken-arbiter restart
```

Créer un nouveau module de type SLA

Pour pouvoir configurer un module de type "s la", il faut faire un nouveau fichier de configuration grâce au fichier d'exemple fourni par défaut.

- Pour commencer, il faut choisir le nom du nouveau module.
 - Pour l'exemple, on l'appelle "Mon-Module-Sla".
 - Remplacer dans l'exemple le mot "Mon-Module-Livedata-Sla-Provider" par le nom qui a été choisi.
- Puis il faut créer le fichier de configuration :
 - Copier le fichier de définition du module d'exemple : `/etc/shinken-user-example/configuration/daemons/brokers/modules/sla/sla-example.cfg` dans le répertoire de définition des modules `/etc/shinken/modules/`.
 - (Exemple : `/etc/shinken/modules/Mon-Module-Sla.cfg`)

```
cp /etc/shinken-user-example/configuration/daemons/brokers/modules/sla/sla-example.cfg /etc/shinken/modules/Mon-Module-Sla.cfg
```

- Ensuite, il faut modifier le fichier nouvellement créé pour configurer le nouveau module.
 - Il faut vérifier que le fichier appartienne à l'utilisateur shinken et qu'il possède le droit d'édition. Si ce n'est pas le cas, il faut effectuer les commandes suivante :

```
chown -R shinken:shinken /etc/shinken/modules/Mon-Module-Sla.cfg
chmod u+w /etc/shinken/modules/Mon-Module-Sla.cfg
```

- On change le nom du module en "**Mon-Module-Sla**" dans le fichier `/etc/shinken/modules/Mon-Module-Sla.cfg`

```
...
# Module name [ Must be unique ]
MANDATORY ]
```

```

#

module_name                               Mon-Module-Sla
...

```

- Ensuite, il faut ajouter le nouveau module dans le module de type "sla" correspondant.
 - Dans notre exemple, on ajoute le module "Mon-Module-Sla" au démon Broker définie dans le fichier /etc/shinken/brokers/broker-master.cfg

```

define broker {
    [...]
    modules          Module 1, Module 2, Module 3, Mon-Module-Sla
    [...]
}

```

- Puis pour finir il faut redémarrer l'Arbiter pour que le Broker puisse prendre en compte ce nouveau module.

Configuration

La configuration du module se trouve par défaut dans le fichier **/etc/shinken/modules/sla.cfg**

- Un exemple dans **/etc/shinken-user-example/configuration/daemons/brokers/modules/sla/sla-example.cfg**

Exemple de fichier de configuration

```

=====
# sla
=====
# Daemons that can load this module:
# - broker (to save sla information into a mongodb database)
# Modules that can load this module:
# - WebUI (to display sla data to the users)
# This module compute and save SLA values into a mongodb database
=====

define module {

    # #
    #     MODULE IDENTITY      #
    # #

    # Module name [ Must be unique ]                                [ MANDATORY ]
    #
    module_name                                sla

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

    # #
    #     MODULE OPTIONS      #
    # #

    # Duration in day to keep SLA info.
    # If value is -1 (kept forever) the MongoDB database will grow endlessly.
    #
    #         Default : -1 => kept forever ( days )
    #         -> Recommended : 547 ( corresponds to 18 months )
    #
    nb_stored_days                                547

    # Time of day when cleanup of SLA is performed
    # When nb_stored_days is set : daily cleanup is done at requested time
    #
    #         Default : 03:02
    #         -> format is HH:MM with
    #         -> HH: are the hour of the day (an integer between 0 and 23)
    #         -> MM: are the minutes          (an integer between 0 and 59)

```

```

#
# time_when_delete_old_SLA                                03:02

# Days to keep raw SLA.
# In case of issue, these data will be used to re-perform SLA computation.
# The drawback of this feature is that it takes more disk space.
#
#         Default : 7 ( days )
#
# keep_raw_sla_day                                         7

# #
#   SLA CALCULATION      #
# #

# Some status can impact
#   -> positively (counted as OK/UP)
#   -> negatively (counted as CRITICAL/DOWN)
#   -> not impact the SLA
# (is not counted, meaning the period of study is reduced by the period that is not counted).
# This configuration aims at giving Shinken administrators a way to configure
# how the SLA are calculated.

# SLA are computed on a daily basis.
# SLA of the current day are always recomputed after a configuration change.
# SLA from days before are by default not recomputed.
#
#         Default : 0 => Disable ( old SLA will not be recalculated )
#         ...      : 1 => Enable ( old SLA will be recomputed with current settings )
#
# recompute_old_sla                                         0

# Warning periods
#
#         Default : 0 => Warning counts as DOWN
#         ...      : 1 => Warning counts as UP
#
# warning_counts_as_ok                                     0

# Unknown periods
#
#         Default : include => "Unknown" status is counted negatively in the SLA.
#         ...      : exclude => "Unknown" are not counted from SLA considered period.
#         ...      : ok      => "Unknown" are considered as UP periods
#
# unknown_period                                           include

# No_data periods ( "Missing data" and "Shinken inactive" status )
#
#         Default : include => Only status is considered. "Missing data" and "Shinken inactive"
#         status are counted negatively in the SLA.
#         ...      : exclude => No_data are not counted from SLA considered period.
#         ...      : ok      => No_data are considered as UP periods.
#
# no_data_period                                           include

# Downtime periods
#
#         Default : include => Only status is considered.
#         ...      : exclude => Downtimes are not counted from SLA considered period.
#         ...      : ok      => Downtimes are considered as UP periods.
#         ...      : critical => Downtimes are considered as DOWN periods.
#
# downtime_period                                          include

# #
#   SLA STORED OUTPUT      #
# #

# This option enables or disables storing sla outputs.
#

```

```

#           Default : 1 => Enable ( the output will be stored )
#           ...      : 0 => Disable ( the output and long output will not be stored
#                               downtime and acknowledge will still be stored )
#
# store_output                                     1

# This option enables or disables storing sla long outputs.
#
#           Default : Enable => 1 ( the long output will be stored )
#           ...      : Disable => 0 ( the long output will not be stored
#                               output, downtime and acknowledge will still be stored )
#
# store_long_output                               1

# This option will be used to filter which outputs and long outputs
# to store depending on the status of the sla.
#
#           Default : empty => ( all output states are stored )
#           ...      : list of status => ( format is State1, State2, ... )
#           -> State ok      :OK
#           -> State warning :WARNING
#           -> State critical:CRITICAL
#           -> State unknown :UNKNOWN
#           Example : OK, UNKNOWN
#
# list_of_stored_output_status

# #
# 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 sla data
#
#           Default : shinken
#
# database                                     shinken

# username/password to authenticate to MongoDB.
# Both parameters must be provided for authentication to function correctly.
#
# database__username

#
# database__password

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

# If the SSH connection goes wrong, then retry use_ssh_retry_failure time before shinken_inactive
#
#           Default : 1 ( number of retry )
#
# 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 ( number of 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 sla processing load among CPUs.
#
#         Default : 1 => X workers
#
# 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

# Maximum number of elements archived in one bulk pass.
# Use this if at 00:05 (archive time) your MongoDB is saturated
#
#         Default : 10 000 ( number of elements )
#
# size_chunk_to_archive                    10000

# Time between two chunk to archive.
# Use this if at 00:05 (archive time) your MongoDB is saturated

```

```

#
#           Default : 0.1 ( seconds )
#
# time_between_two_chunks                                0.1

# Max number of sla remove each daily_clean_pause_time.
# Use if nb_stored_days is not -1. ( Daily clean time is activated )
# Use this if at 03:02 (daily clean time) your MongoDB is saturated.
#
#           Default : 10 000 ( number of sla )
#
# daily_clean_batch_size                                10000

# Delay between 2 sla clean.
# Use if nb_stored_days is not -1. ( Daily clean time is activated )
# Use this if at 03:02 (daily clean time) your MongoDB is saturated.
#
#           Default : 2 ( seconds )
#
# daily_clean_pause_time                                2

# Max number of sla archive migrate save at same time.
# Use this if after an Shinken update your MongoDB is saturated.
#
#           Default : 1 000 ( sla )
#
# broker_module_sla_migration_batch_size                1000

# Delay between 2 migrating batch save.
# Use this if after an Shinken update your MongoDB is saturated.
#
#           Default : 0.1 ( seconds )
#
# broker_module_sla_migration_pause_time                0.1

# Split historical sla_archive collection in daily archive collections
# As this may require extra disk space to run, you can disable it here in order to delay it until
# more disk space is available.
# After completion, performance and disk space management will be greatly improved
#
#           Default : 1 => Enable
#           ...      : 0 => Disable
#
# broker__module_sla__enable_migration_sla_archive_in_daily_collections 1
}

```

Détails des sections composant le fichier de configuration

Identification du module

Il est possible de définir plusieurs instances de module de type "sla" dans l'architecture Shinken.

- Chaque instance devra avoir un nom unique.

Nom	Type	Unité	Défaut	Description
module_name	Texte	---	sla	Shinken conseille de choisir un nom en fonction de l'utilisation du module pour que la configuration soit simple à maintenir. Doit être unique.
module_type	Texte	---	sla	Ne peut être modifié.

Suppression des anciennes entrées dans la base d'archives du module SLA

```
# Duration in day to keep SLA info.
# If value is kept forever the MongoDB database will grow endlessly.
#
#      Default : -1 => kept forever ( days )
#      -> Recommended : 547 ( corresponds to 18 months )
#
# nb_stored_days                                547

# Time of day when cleanup of SLA is performed
# When nb_stored_days is set : daily cleanup is done at requested time
#
#      Default : 03:02
#      -> format is HH:MM with
#      -> HH: are the hour of the day (an integer between 0 and 23)
#      -> MM: are the minutes          (an integer between 0 and 59)
#
# time_when_delete_old_SLA                      03:02
```

Les entrées dans la base d'archives du module SLA sont supprimées toutes les 24h.

Nom	Type	Unité	Défaut	Description
nb_stored_days	Entier	jours	-1	Détermine le nombre de jours à garder dans la base d'archives du module SLA. La valeur minimale acceptée correspond à 7 jours. La valeur -1 signifie qu'on veut garder toutes les entrées dans la base d'archives du module SLA, et il n'y a pas de suppression quotidienne.
time_when_delete_old_SLA	Texte	heures	03:02	Heure de la journée à laquelle les entrées dans la base d'archives du module SLA seront supprimées. Les données gardées correspondent aux jours définis par la valeur de la propriété nb_stored_days



Si le Broker est éteint et que la suppression n'est pas faite depuis plus de 24 h, elle se fera automatiquement au démarrage du Broker.



S'il n'y a aucune trace du dernier nettoyage ou que le dernier nettoyage date de plus de 24 heures, le nettoyage de la base d'archives du module SLA s'exécutera de nouveau.

Sauvegarde des SLA brut

```
# Days to keep raw SLA.
# In case of issue, these data will be used to re-perform SLA computation.
# The drawback of this feature is that it takes more disk space.
#
#      Default : 7 ( days )
#
# keep_raw_sla_day                                7
```

Afin de gérer le flux de données, on enregistre sous un format brut (*raw_sla*) les informations de SLA. Toutes les nuits, ces données sont agrégées dans un nouveau format (*sla_archive*) pour accélérer la lecture et réduire l'espace de stockage nécessaire.

S'il y a une erreur lors du passage entre les deux formats, des données peuvent être irrémédiablement perdues. Afin d'éviter de perdre des données lorsque survient ce type d'erreur, il est possible de conserver les données brutes quelque temps.

Le paramètre "keep_raw_sla_day" permet de choisir combien de temps garder ces données.

Il est possible de diminuer ce paramètre si on manque d'espace disque et que les données SLA ne sont pas primordiales. Inversement, si les données SLA sont critiques, il est possible d'augmenter ce nombre, ce qui permettra de limiter la perte de données SLA.

Nom	Type	Unité	Défaut	Description
keep_raw_sla_day	Entier	jours	7	Nombre de jours durant lesquels sont gardées les données brutes.



La récupération n'est possible qu'avec l'aide du support dédié

Option de calcul du taux de SLA

```
# #
# SLA CALCULATION #
# #

# Some status can impact
#   -> positively (counted as OK/UP)
#   -> negatively (counted as CRITICAL/DOWN)
#   -> not impact the SLA
# (is not counted, meaning the period of study is reduced by the period that is not counted).
# This configuration aims at giving Shinken administrators a way to configure
# how the SLA are calculated.

# SLA are computed on a daily basis.
# SLA of the current day are always recomputed after a configuration change.
# SLA from days before are by default not recomputed.
#
#   Default : 0 => Disable ( old SLA will not be recalculated )
#   ...      : 1 => Enable ( old SLA will be recomputed with current settings )
#
# recompute_old_sla                                0

# Warning periods
#
#   Default : 0 => Warning counts as DOWN
#   ...      : 1 => Warning counts as UP
#
# warning_counts_as_ok                              0

# Unknown periods
#
#   Default : include => "Unknown" status is counted negatively in the SLA.
#   ...      : exclude => "Unknown" are not counted from SLA considered period.
#   ...      : ok      => "Unknown" are considered as UP periods
#
# unknown_period                                    include

# No_data periods ( "Missing data" and "Shinken inactive" status )
#
#   Default : include => Only status is considered. "Missing data" and "Shinken inactive"
#   status are counted negatively in the SLA.
#   ...      : exclude => No_data are not counted from SLA considered period.
#   ...      : ok      => No_data are considered as UP periods.
#
# no_data_period                                    include

# Downtime periods
#
#   Default : include => Only status is considered.
#   ...      : exclude => Downtimes are not counted from SLA considered period.
#   ...      : ok      => Downtimes are considered as UP periods.
#   ...      : critical => Downtimes are considered as DOWN periods.
#
# downtime_period                                    include
```

Nom	Type	Unité	Défaut	Description
recompute_old_sla	Booléen	---	0	• 1 : Tous les taux de disponibilité changeront à chaque changement de configuration. • 0 : Les SLA déjà archivés ne changerons pas.

warning_counts_as_ok	Booléen	---	0	• 1 : On considère que le service est toujours rendu même de manière potentiellement dégradée. • 0 : On considère que si le service n'est pas rendu de manière optimale, il n'est pas bien rendu et donc fait baisser le taux de disponibilité.
unknown_period	String	---	include	▪ ok : On considère que le service a donné un état, ainsi le service est encore rendu même de manière potentiellement dégradée. ▪ include : On considère que si l'on ne peut savoir si le service est rendu, c'est qu'il n'est pas rendu. ▪ exclude : On considère l'état trop imprécis pour modifier le taux de disponibilité.
no_data_period	String	---	include	• ok : On considère que même si la supervision ne l'a pas confirmé, le service est rendu. • include : On considère que si l'on ne peut pas savoir si le service est rendu, c'est qu'il n'est pas rendu. • exclude : On considère que l'état de la supervision (<i>de Shinken</i>) n'impacte pas le taux de disponibilité.
downtime_period	String	---	include	• ok : On considère que les périodes de maintenance planifiées font partie du service, donc le service est rendu. • critical : On considère que lors d'une maintenance planifiée, le service n'est plus rendu. • include : On considère que la maintenance est planifiée et ainsi que cette période n'impacte pas le service. • exclude : On considère que seul l'état du service compte pour le taux de disponibilité.



Plus de détails sur ces paramètres et sur le fonctionnement des SLA sur cette page : [Calcul du taux de SLA](#)

Option de stockage des Resultats et Resultats longs

```
# #
#   SLA STORED OUTPUT   #
# #

# This option enables or disables storing sla outputs.
#
#   Default : 1 => Enable ( the output will be stored )
#   ...      : 0 => Disable ( the output and long output will not be stored
#                                   downtime and acknowledge will still be stored )
#
# store_output                                     1

# This option enables or disables storing sla long outputs.
#
#   Default : Enable => 1 ( the long output will be stored )
#   ...      : Disable => 0 ( the long output will not be stored
#                                   output, downtime and acknowledge will still be stored )
#
# store_long_output                                1

# This option will be used to filter which outputs and long outputs
# to store depending on the status of the sla.
#
#   Default : empty => ( all output states are stored )
#   ...      : list of status => ( format is State1, State2, ... )
#   -> State ok      :OK
#   -> State warning :WARNING
#   -> State critical:CRITICAL
#   -> State unknown :UNKNOWN
#   Example : OK, UNKNOWN
#
# list_of_stored_output_status
```

Afin de limiter l'espace pris par la base des SLA, il est possible de filtrer les résultats et les résultats longs sauvegardés dans la base.

Il est possible de monitorer l'espace pris par la base grâce au modèle d'hôte (voir la page [Modèle shinken-broker-module-sla-writer](#)).

Les résultats et les résultats longs des sondes ne sont que les textes donnés par la commande de vérification.

Exemple : PING CRITICAL - Packet loss = 100%

Nom	Type	Unité	Défaut	Description
store_output	Booléen	---	1	1 : Les résultats des sondes seront sauvegardés. 0 : Les résultats des sondes ne seront pas sauvegardés.
store_long_output	Booléen	---	1	1 : Les résultats longs des sondes seront sauvegardés. 0 : Les résultats longs des sondes ne seront pas sauvegardés.
list_of_stored_output_status	Liste	---		Cette option sera utilisée pour filtrer les résultats et les résultats longs qui seront stockés en fonction du statut de la vérification. Les éléments de la liste sont séparés par une virgule. Si la valeur est laissée à vide alors tous les statuts sont pris en compte. Les statuts possibles sont : - OK - WARNING - CRITICAL - UNKNOWN



Shinken conseille pour limiter l'espace pris par la base des SLA. Il est possible de limiter le nombre de jours sauvegardés via l'option "nb_stored_days" plutôt que de ne pas sauvegarder les résultats et les résultats longs très utiles lors de l'analyse d'incident.

Accès à la base MongoDB

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

Pour se connecter à la base MongoDB utilisée 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

Configuration de l'URI de connexion et de l'authentification par mot de passe

```
# #
# 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 sla data
#
# Default : shinken
#
# database shinken

# username/password to authenticate to MongoDB.
```

```
# Both parameters must be provided for authentication to function correctly.
#
# database__username

#
# database__password
```

Nom	Type	Unité	Défaut	Description
uri	Texte	URL	mongodb://localhost/?w=1&fsync=false	La syntaxe de l'uri de MongoDB est disponible à l'adresse https://docs.mongodb.com/manual/reference/connection-string/ .
database	Texte	---	shinken	Nom de la base de données où sont stockées les données SLA.
database__username	Texte	---		Utilisateur pour l'authentification avec mot de passe à la base MongoDB. Utile uniquement si l'activation par mot de passe a été activé (voir la page MongoDB - activation de l'authentification par mot de passe)
database__password	Texte	---		Mot de passe de l'utilisateur utilisé pour l'authentification avec mot de passe à la base MongoDB. Utile uniquement si l'activation par mot de passe a été activé (voir la page MongoDB - activation de l'authentification par mot de passe)

Connexion directe au serveur MongoDB

Par défaut, le module se connecte de manière directe à la base MongoDB, définie avec les paramètres communs listés ci-dessus, car le paramètre "use_ssh_tunnel" est à 0.

Connexion par SSH 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 la page [Sécurisation des connexions aux bases MongoDB](#)), mais bien plus complexe à mettre en place.

La méthode de connexion par SSH est ainsi préférable pour des raisons pratiques et de sécurité.

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

# If the SSH connection goes wrong, then retry use_ssh_retry_failure time before shinken_inactive
#
#      Default : 1 ( number of retry )
#
# 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*), il faut 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	Description
use_ssh_tunnel	Booléen	---	0	• 1 : Connexion par tunnel SSH. • 0 : Connexion directe.
use_ssh_retry_failure	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.
ssh_user	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 utilisée pour établir le tunnel.
ssh_tunnel_timeout	Entier	secondes	10	Spécifie le timeout en secondes de la vérification du tunnel SSH avant que la connexion vers MongoDB soit effectuée.

Pour configurer les clés SSH à utiliser, voir la page [Création automatique et gestion de la clé SSH de l'utilisateur shinken](#)

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 ( number of 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.

Élection : processus de MongoDB pour choisir un nouveau membre Primaire si le membre Primaire devient inaccessible.

(Voir la page [Haute disponibilité de la base MongoDB \(mise en place d'un cluster\)](#))

Dans le cas de l'utilisation d'un cluster MongoDB, lorsque le membre Primaire devient inaccessible, une nouvelle élection est déclenchée, 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 égal 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	Description
auto_reconnect_max_try	Entier	essais	4	Nombre d'essais de reconnexion à la base.
auto_reconnect_sleep_between_try	Entier	secondes	3	Temps entre chaque essai en secondes.

Les valeurs par défauts du fichier laissent 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 SLA

```
# #
# 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 sla processing load among CPUs.
#
# Default : 1 => X workers
#
# broker_module_nb_workers 1
```

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

Il est possible de changer ce paramètre si l'utilisation CPU du processus : "NOM DU BROKER [- Module: sla][Worker: 0]" est trop élevé.

Nom	Type	Unité	Défaut	Description
broker_module_nb_workers	Entier	nombre de worker	1	Nombre de workers (<i>nombre de clone du module</i>) qui traitent le flux de données pour sauvegarder les données SLA dans la base MongoDB.



Ne pas dépasser le nombre de core cpu de la machine : cela serait contre-productif pour les performances.

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

# Maximum number of elements archived in one bulk pass.
# Use this if at 00:05 (archive time) your MongoDB is saturated
#
#     Default : 10 000 ( number of elements )
#
# size_chunk_to_archive                      10000

# Time between two chunk to archive.
# Use this if at 00:05 (archive time) your MongoDB is saturated
#
time_before_shinken_inactive #           Default : 0.1 ( seconds
)
#
# time_between_two_chunks                    0.1

# Max number of sla remove each daily_clean_pause_time.
# Use if nb_stored_days is not -1. ( Daily clean time is activated )
# Use this if at 03:02 (daily clean time) your MongoDB is saturated.
#
#     Default : 10 000 ( number of sla )
#
# daily_clean_batch_size                    10000

# Delay between 2 sla clean.
# Use if nb_stored_days is not -1. ( Daily clean time is activated )
# Use this if at 03:02 (daily clean time) your MongoDB is saturated.
#
#     Default : 2 ( seconds )
#
# daily_clean_pause_time                    2

# Max number of sla archive migrate save at same time.
# Use this if after an Shinken update your MongoDB is saturated.
#
#     Default : 1 000 ( sla )
#
# broker_module_sla_migration_batch_size    1000
```

```
# Delay between 2 migrating batch save.
# Use this if after an Shinken update your MongoDB is saturated.
#
#           Default : 0.1 ( seconds )
#
# broker_module_sla_migration_pause_time           0.1

# Split historical sla_archive collection in daily archive collections
# As this may require extra disk space to run, you can disable it here in order to delay it until
# more disk space is available.
# After completion, performance and disk space management will be greatly improved
#
#           Default : 1 => Enable
#           ...      : 0 => Disable
#
# broker__module_sla__enable_migration_sla_archive_in_daily_collections 1
```



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

Nom	Type	Unité	Défaut	Description
time_before_shinken_inactive	Entier	secondes	30	Temps d'inactivité du Broker avant de considérer que Shinken est inactif. Utiliser cette option si le temps de boucle du Broker dépasse 30 secondes.
size_chunk_to_archive	Entier	nombre d'éléments	10000	Nombre maximum d'éléments archivés en un seul passage en masse. Utiliser cette option si à 00:05 (<i>heure d'archivage</i>) le MongoDB est saturée.
time_between_two_chunks	Décimal	secondes	0.1	Temps entre deux lots d'éléments à archiver. Utiliser cette option si à 00:05 (<i>heure d'archivage</i>) le MongoDB est saturé.
daily_clean_batch_size	Entier	nombre d'archives SLA	10000	Nombre maximum d'archive SLA supprimée chaque daily_clean_pause_time . Fonctionne que si le paramètre nb_stored_days n'est pas -1. (<i>L'heure de nettoyage quotidien est activée</i>) Utiliser cette option si à 03:02 (<i>heure de nettoyage quotidien</i>) le MongoDB est saturé.
daily_clean_pause_time	Entier	secondes	2	Délai entre deux nettoyages des données SLA. Fonctionne si le paramètre nb_stored_days n'est pas -1. (<i>Qu'il y a une suppression journalière des archives SLA</i>) Utiliser cette option si à 03:02 (<i>heure de nettoyage quotidien</i>) la base MongoDB est saturée.
broker_module_sla_migration_batch_size	Entier	nombre de données SLA	1000	Nombre maximum d'archives SLA migrées et sauvegardées en même temps. Utiliser cette option si après une mise à jour de Shinken le MongoDB est saturé.
broker_module_sla_migration_pause_time	Décimal	secondes	0.1	Délai entre deux sauvegardes de lots de migration. Utiliser cette option si après une mise à jour de Shinken le MongoDB est saturé.
broker__module_sla__enable_migration_sla_archive_in_daily_collections	Booléen	---	1	Si la collection sla_archive est présente, autorise le processus de migration à la découper en plusieurs collections, pour regrouper les archives par jour. Pendant l'opération, qui peut durer plusieurs semaines suivant la quantité de données présentes dans sla_archive , un supplément d'espace disque va être nécessaire : - durant que les données d'une journée sont copiées dans la nouvelle collection, elles vont être présentes deux fois sur disque - après la suppression de données dans sla_archive, tant que cette collection existe, MongoDB ne restitue pas systématiquement l'espace libéré au système. En effet, il attend que la quantité de données libérées dépasse certains volumes pour effectivement retourner cet espace libre au système. Après l'opération, les temps d'accès aux archives pour consultation ou nettoyage seront grandement améliorés. La suppression d'une journée d'archive s'accompagnera systématiquement de la libération de l'espace disque associé.

Erreurs dans le Module

Afin de ne pas casser la base et les données de SLA, si le module à une erreur inattendue comme un crash, alors le module s'arrête et n'est pas automatiquement redémarré.

Lancer la commande shinken-healthcheck pour trouver l'erreur Fatale

```
Module sla:  
  ERROR: Your module have a FATAL error and it will NOT be restart to protect your data.  
  ERROR: You must contact your dedicated support with your log and shinken-backup.  
  ERROR: Fatal error cause by : error for testing
```