

# Module SLA

## Explication

Le module SLA 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. Il est également possible, via ce fichier, de modifier la méthode de calcul des SLA (par exemple, choisir de considérer un Warning comme une période positive de SLA, ou encore d'exclure les périodes de maintenance dans le calcul).

## Mécanisme des données manquantes / Shinken Inactif



Voir : [Détermination de l'état de l'hôte](#) ( pour plus d'information sur les cas particuliers )

Lorsque Le Broker n'est pas démarré les hôtes et les checks prennent le statut Shinken Inactif.

Ce statut est calculé à partir du *pouls* (*heartbeat*) du Broker c'est à dire à chaque boucle du Broker on écrit un battement dans la base si il n'y a plus de battement pendant un période on considère que le Broker n'est plus actif.

## Configuration

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

```
#####
# 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 {

# 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-d05cd3505adb11e5884b080027f08538
    _SE_UUID_HASH     05d3d1d1ccef1f5e03b43936aad25e68f
# End of Shinken Enterprise part

#==== Module identity =====
# Module name. Must be unique
module_name          sla

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

#==== 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/?w=1&fsync=false

# 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

# SSH Timeout used to test if the SSH tunnel is viable or not, in seconds
```

```

# Default: 5
# ssh_tunnel_timeout      5

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

# Which database is used to store sla data
database          shinken

##### Module options #####
# Raw SLA can be kept during X days. 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.
#keep_raw_sla_day  7          ;optional, defaults to 7

# Duration in day to keep SLA info,
# Default value is -1. It mean SLA are keep forever, in this case to mongo database will grow endlessly.
# Minimal value is 7 day
#nb_stored_days   -1

# Time of day the SLA archive cleanup is performed
# Default value, if unset, is 03:02
# Daily cleanup is done at requested time when nb_stored_days is set
# format is HH:MM with
# - HH is 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

# 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.
# If 1, old SLA will be recomputed with current settings.
# If 0, old SLA will not be recalculated [default]
# recompute_old_sla      0

##### SLA calculation #####
# Some status can impact positively (counted as OK/UP), negatively (counted as CRITICAL/DOWN) or 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.

# If 1, Warning counts as UP
# If 0, Warning counts as DOWN [default]
# warning_counts_as_ok   0

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

# == Downtime periods ==
# - include: Only status is considered. [default]
# - 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

##### INTERNAL options #####
#INTERNAL : DO NOT EDIT FOLLOWING PARAMETRE WITHOUT YOUR DEDICATED SUPPORT
# == time of inactivation of the broker before considering that shinken is inactive (in sec) ==
#time_before_shinken_inactive  30
# == maximum number of elements archived in one bulk pass ==
#size_chunk_to_archive         10 000

```

```
# == time between two chunk to archive ==
#time_between_two_chunks      0.1
# == default value of the interval check (in minutes) ==
#default_check_interval       5
# == delay before the creation of missing data period (in check intervale) ==
#margin_create_new_range      1.5
# == max delay before creating missing data period (in minutes) ==
#margin_create_new_range_max  10

# Explanatory example of the property margin_create_new_range
# For an element with a check interval at 1min and margin_create_new_range at 1.5 which equals 1min30s
of time delay.
# If the interval check is at 1h the delay would be at 1h30 but the delay is limited by
margin_create_new_range_max which limits the delay to 10min.
#
# An OK status is given by the scheduler at 12h30
# A new OK status is given by the scheduler at 12h40
# The scheduler should have given a new status at 12h31 but it gave it at 12h40 which is 9min of time
delay.
# So that 9min > 1min30s a missing data period is created.

}
```

## Configurer l'accès à la base MongoDB

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

```

define module{
...
...
...
##### 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/?w=1&fsync=false

# 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

# SSH Timeout used to test if the SSH tunnel is viable or not, in seconds
# Default: 5
# ssh_tunnel_timeout  5

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

# Which database is used to store sla data
database            shinken

...
...
...
}

```

Les données SLA sont stockées dans la base Mongo locale au Broker

Pour se connecter au serveur Mongo utilisé pour le stockage des données SLA, 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 SLA se connecte de manière directe au serveur Mongo pour y lire et écrire les données SLA.

Dans la configuration du module SLA, 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    sla
...
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 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 donc préférable pour des raisons pratiques et de sécurité.

## Connexion par SSH au serveur Mongo

Le module SLA 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
use_ssh_tunnel	Activation du tunnel ou non (0 ou 1 ). Par défaut, il n'est pas activé ( à 1 )
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
ssh_user	L'utilisateur avec lequel le tunnel sera établi
ssh_keyfile	une clé ssh privée présent sur le serveur Shinken (par défaut ~/.ssh/id_rsa) qui sera utilisé pour établir le tunnel.
ssh_tunnel_timeout	Spécifie le timeout de la vérification du tunnel SSH avant que la connexion vers mongo soit effectuée

nofollow

- 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.

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

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

Nom du paramètre	Description	Valeur par défaut
nb_stored_days	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.	-1
time_when_delete_old_SLA	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 <b>nb_stored_days</b>	03:02



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 de la base d'archives du module SLA, le nettoyage s'exécutera.