

CPU Stats by SSH

Sommaire

Contexte

Paramétrage

- Données utilisées provenant des modèles
 - Données communes pour les checks des modèles
 - Authentification
 - Données spécifiques pour ce check
 - Données DFE (Duplicate Foreach)
- Données utilisées provenant du check

Résultat

- Exemple
- Interprétation des données
 - Statut
 - Résultat
 - Résultat Long

Métriques

- Définition
- Exemple

Les Erreurs

- Erreurs spécifiques à ce check
 - MONITORED HOST - BAD STATE – Command 'mpstat' not found.
- Erreurs de connexion (communes à tous les checks)
 - UNKNOWN – Username/PublicKey combination invalid
 - UNKNOWN – Unable to extract public key from private key file : Unable to open private key file
 - UNKNOWN – Unable to extract public key from private key file : Wrong passphrase or invalid/unrecognized private key file format
 - UNKNOWN – Connection refused (os error 111)
 - UNKNOWN – Name or service not known

Détails de l'algorithme

Description

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.
- 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).

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

Vous trouverez une erreur FATAL avec la commande shinken-healthcheck.

? Unknown Attachment

Activation du module

Le module `sla` est un module qui peut être activé seulement sur le démon Broker.

- L'activation du module s'effectue en ajoutant le nom de ce module dans le fichier de configuration du démon Broker.
- Pour ce faire, ouvrir le fichier de configuration du Broker à l'emplacement **/etc/shinken/modules/**, et ajouter le nom de votre module "sla".

Exemple: par défaut, nous livrons un module dont le nom est "sla":

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

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

```
service shinken-arbiter restart
```

Configuration

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

- Vous trouverez aussi systématiquement un exemple dans `/etc/shinken-user-example/configuration/daemons/brokers/modules/sla/sla.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 kept forever the MongoDB database will grow endlessly.
    # >>> Default : -1 ( kept forever )
    # >>> 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
    # 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)
    # >>> Default : 03:02
    # 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.
#   Enable : 1 ( old SLA will be recomputed with current settings )
#   >>> Disable : 0 ( old SLA will not be recalculated ) ( Default )
# recompute_old_sla                                0

# Warning periods
#   1 : Warning counts as UP
#   >>> 0 : Warning counts as DOWN ( Default )
# warning_counts_as_ok                              0

# Unknown periods
#   >>> include : "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

# #
#   SLA STORED OUTPUT      #
# #

# This option enables or disables storing sla outputs.
# >>> Enable : 1 ( the output will be stored ) ( Default )
# Disable : 0 ( 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.
# >>> Enable : 1 ( the long output will be stored ) ( Default )
# 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.
# format is State1, State2, ...
#   -> State ok      : OK
#   -> State warning : WARNING
#   -> State critical : CRITICAL
#   -> State unknown : UNKNOWN
# >>> All states      : empty ( all output states are stored ) ( Default )
# 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

# 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
# >>> Enable : 1 ( with ssh tunnel ) ( Default )
# Disable : 0 ( direct connection )
# 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 sla processing load among CPUs.
# >>> Default : 1 ( one 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

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

```

```

# size_chunk_to_archive                                10000

# Time between two chunk to archive.
# Use this if at 00:05 (archive time) your MongoDB is saturate
# >>> 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 saturate.
# >>> Default : 10 000 ( sla numbers )
# 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 saturate.
# >>> 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 saturate.
# >>> 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 saturate.
# >>> Default : 0.1 ( seconds )
# broker_module_sla_migration_pause_time              0.1

}

```

Détails des sections composant le fichier de configuration

Identification du module

Il est possible de définir plusieurs instances de module de type sla. Chaque instance devra avoir un nom unique.

nom	type	Unités	défaut	commentaire
module_name	Texte	---	sla	
module_type	Texte	---	sla	

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 )
# >>> 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
# 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)
# >>> Default : 03:02
# time_when_delete_old_SLA                           03:02

```

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

nom	type	Unités	défaut	commentaire
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 de la base d'archives du module SLA, le nettoyage s'exécutera.**

Sauvegarde des sla brut


Afin de gérer le flux de donnée, 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 vous manquez d'espace disque et que les données SLA ne sont pas primordiales pour vous. Inversement si pour vous les données SLA sont critiques il est possible d'augmenter ce nombre ce qui permettra de limiter la perte de donnée SLA.

nom	type	Unités	défaut	commentaire
keep_raw_sla_day	Entier	jours	7	Nombre de jours durant lesquels sont gardées les données brut.

 La récupération est possible qu'avec l'aide de votre support dédié

Option de calcul du taux de SLA

La description de ces paramètres ainsi que leurs implications est disponible sur cette page : [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.
# Enable : 1 ( old SLA will be recomputed with current settings )
# >>> Disable : 0 ( old SLA will not be recalculated ) ( Default )
# recompute_old_sla                                0

# Warning periods
# Warning counts as UP : 1
# >>> Warning counts as DOWN : 0 ( Default )
# warning_counts_as_ok                             0

# Unknown periods
# >>> include : "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

```

Option de stockage des Resultats et Resultats longs

```

# #
#     SLA STORED OUTPUT     #
# #

# This option enables or disables storing sla outputs.
# >>> Enable : 1 ( the output will be stored ) ( Default )
# Disable : 0 ( 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.
# >>> Enable : 1 ( the long output will be stored ) ( Default )
# 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.
# format is State1, State2, ...
# -> State ok      : OK
# -> State warning : WARNING
# -> State critical : CRITICAL
# -> State unknown : UNKNOWN
# >>> All states   : empty ( all output states are stored ) ( Default )
# Example: OK, UNKNOWN
# list_of_stored_output_status

```

Option pas décrites

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

Connexion directe au serveur Mongo

```

# 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

```

Connexion par SSH 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 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.

```

# 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
# >>> Enable : 1 ( with ssh tunnel ) ( Default )
# Disable : 0 ( direct connection )
# 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

```

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	Unités	défaut	commentaire
use_ssh_tunnel	Booléen		0	<ul style="list-style-type: none"> • 1 : Connection par tunnel SSH • 0 : Connection direct
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 utilisé 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

Le tunnel SSH va permettre au module de se connecter comme si ses requêtes étaient locales 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

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 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 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 mongo.

nom	type	Unités	défaut	commentaire
-----	------	--------	--------	-------------

<pre>auto_reconnect_max_try</pre>	Entier	essais	4	Nombre d'essais de reconnexion à la base
<pre>auto_reconnect_sleep_between_try</pre>	Entier	secondes	3	Temps entre chaque essais en seconde

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 ( one worker )
# 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	Unités	défaut	commentaire
<pre>broker_module_nb_workers</pre>	Entier	nombre de worker	1	Nombre de workers (nombre de clone du module) qui traitent le flux de donnée 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.