

Memory SNMP (linux)

Sommaire

- Contexte
- Exemple
 - Exemple de résultat
- Données et métriques
 - Données
 - Métriques

Description

Le module de type `livedata_module_sla_provider` est un module qui permet au module `broker-module-livedata` du Broker de fournir une API de consultation des données SLA archivées.

- `/api/v1/sla`
- Il récupère les données SLA présentes dans la base de données Shinken, les données SLA sont calculées à la fin de la journée, donc **la dernière donnée disponible est celle d'hier**.

Pour plus de détails sur cette API, vous pouvez consulter sa [documentation d'utilisation](#).

Activation du module

Activer le module `livedata-module-sla-provider` livré par défaut

Par défaut, l'installation ou la mise à jour de Shinken Entreprise va mettre à disposition une définition du module de type `livedata_module_sla_provider`

- La configuration de ce module se trouve par défaut dans le fichier : `/etc/shinken/modules/livedata-module-sla-provider.cfg`
- Vérifier que le nom "livedata-module-sla-provider" est présent dans le paramètre **modules** du fichier `/etc/shinken/modules/broker-module-livedata.cfg` (*ou le .cfg que vous utilisez pour définir les options du broker-module-livedata*).
 - Ce module est normalement présent dans la liste des modules livrés par défaut.
 - S'il n'y est pas (*car modifié*), vous pouvez le rajouter pour qu'il soit de nouveau actif.
Exemple:

```
define module {
    [...]

    module_name          broker-module-livedata

    [...]

    modules              Module 1, Module 2, Module 3, livedata-module-sla-provider
    [...]
}
```

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

```
service shinken-arbiter restart
```



Il ne peut y avoir qu'un seul module de type `livedata_module_sla_provider` par module `broker-module-livedata`.



Si vous avez plusieurs modules `broker-module-livedata` présents dans votre architecture, n'oubliez pas d'activer le module de type `livedata_module_sla_provider` dans la configuration de chacune d'elles.

Définir vos propres modules de type `livedata_module_sla_provider`

Pour pouvoir définir ce module selon vos besoins, il vous sera possible de définir votre module grâce au module d'exemple fourni par défaut.

Pour configurer votre module de type `livedata_module_sla_provider`, commencez par choisir un nom à lui donner.

- Pour l'exemple, nous allons l'appeler "Mon-Module-Livedata-Sla-Provider".
- Remplacer dans l'exemple le mot "livedata-module-sla-provider" par le nom que vous aurez choisi.

Pour définir votre module à partir du module fourni par défaut, vous devez :

- Copier le fichier de définition du module d'exemple : `/etc/shinken-user-example/configuration/daemons/brokers/modules/broker-module-livedata/modules/livedata-module-sla-provider/livedata-module-sla-provider-example.cfg` dans le répertoire de définition des modules `/etc/shinken/modules/` et modifier son nom pour lui donner le nom que vous avez choisi.

Exemple : `/etc/shinken/modules/livedata-module-sla-provider__Mon-Module-Livedata-Sla-Provider.cfg`

```
cp /etc/shinken-user-example/configuration/daemons/brokers/modules/broker-module-livedata/modules
/livedata-module-sla-provider/livedata-module-sla-provider-example.cfg /etc/shinken/modules/livedata-
module-sla-provider__Mon-Module-Livedata-Sla-Provider.cfg
```

- Dans ce nouveau fichier, changez le nom du module :

```
...
# Module name [ Must be unique ]
MANDATORY ]
#
module_name Mon-Module-Livedata-Sla-Provider
...
```

- Une fois que le fichier a été édité, vérifiez que le fichier possède comme droits utilisateurs shinken. Si ce n'est pas le cas, effectuez la commande suivante :

```
chown -R shinken:shinken /etc/shinken/modules/livedata-module-sla-provider__Mon-Module-Livedata-Sla-
Provider.cfg
```

- Ajouter le nom de votre module de type `livedata_module_sla_provider` au module `broker-module-livedata` en modifiant le paramètre **modules** du fichier `/etc/shinken/modules/broker-module-livedata.cfg`.

```
define module {
    [...]
    modules Module 1, Module 2, Module 3, Mon-
Module-Livedata-Sla-Provider
    [...]
}
```

- Redémarrez l'Arbiter pour que le Broker puisse prendre en compte ce nouveau module.

```
service shinken-arbiter restart
```

Configuration

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

- Vous trouverez aussi systématiquement un exemple dans `/etc/shinken-user-example/configuration/daemons/brokers/modules/broker-module-livedata/modules/livedata-module-sla-provider/livedata-module-sla-provider-example.cfg`

Exemple de fichier de configuration

```
#####
# livedata_module_sla_provider
#####
# Modules that can load this module:
# - broker-module-livedata (to add new route to this module)
# This module is an API providing information on SLA of monitored elements
```

```

#=====
define module {
    # #
    #     MODULE IDENTITY     #
    # #

    # Module name [ Must be unique ]                               [ MANDATORY ]
    #
    module_name                                livedata-module-sla-provider

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

    # #
    #     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
    #
    # livedata_module_sla_provider__database__uri                mongodb://localhost/?w=1&fsync=false

    # Which database contains sla data
    #
    #     Default : shinken
    #
    # livedata_module_sla_provider__database__name                shinken

    # 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 )
    #
    # livedata_module_sla_provider__use_ssh_tunnel                0

    # SSH user to connect to the mongodb server.
    #
    #     Default : shinken
    #
    # livedata_module_sla_provider__ssh_user                      shinken

    # SSH keyfile to connect to the mongodb server.
    #
    #     Default : ~shinken/.ssh/id_rsa
    #
    # livedata_module_sla_provider__ssh_keyfile                  ~shinken/.ssh/id_rsa

    # SSH Timeout used to test if the SSH tunnel is viable or not, in seconds.
    #
    #     Default : 10 ( seconds )
    #
    # livedata_module_sla_provider__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 )
    #
    # livedata_module_sla_provider__database__retry_connection_X_times_before_considering_an_error 4

```

```

# Time between each try
#
#         Default : 3 ( seconds )
#
# livedata_module_sla_provider__database__wait_X_seconds_before_reconnect 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
# retry_connection_X_times_before_considering_an_error * wait_X_seconds_before_reconnect
# must be higher than heartbeatTimeoutSecs in the rs.conf(); of your MongoDB replica set.

# #
# SLA CALCULATION      #
# #

# 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.
#
# livedata_module_sla_provider__no_data_period      include
}

```

Détails des sections composant le fichier de configuration

Identification du module

Il est possible de définir plusieurs instances de module de type `livedata-module-sla-provider` dans votre architecture Shinken.

- Chaque instance devra avoir un nom unique.

Nom	Type	Unité	Défaut	Commentaire
<code>module_ name</code>	Texte	---	<code>livedata-module-sla- provider</code>	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.
<code>module_ type</code>	Texte	---	<code>livedata_module_sla_ _provider</code>	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 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 des paramètres communs aux deux méthodes

```

# 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
#
# livedata_module_sla_provider__database__uri          mongodb://localhost/?w=1&fsync=false

# Which database contains sla data
#
#         Default : shinken
#
# livedata_module_sla_provider__database__name        shinken

```

Nom	Type	Unité	Défaut	Commentaire
livedata_module_sla_provider__database__uri	Texte	URL	mongodb://localhost/?w=1&fsync=false	URL mongo de connexion à la base de données cf. https://docs.mongodb.com/manual/reference/connection-string/
livedata_module_sla_provider__database__name	Texte	---	shinken	Nom de la base de données où sont stockés les données SLA

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 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
#
#      Default : 0 => Disable ( disable ssh tunnel )
#      ...      : 1 => Enable  ( enable ssh tunnel )
#
# livedata_module_sla_provider__use_ssh_tunnel          0

# SSH user to connect to the mongodb server.
#
#      Default : shinken
#
# livedata_module_sla_provider__ssh_user                shinken

# SSH keyfile to connect to the mongodb server.
#
#      Default : ~shinken/.ssh/id_rsa
#
# livedata_module_sla_provider__ssh_keyfile            ~shinken/.ssh/id_rsa

# SSH Timeout used to test if the SSH tunnel is viable or not, in seconds.
#
#      Default : 10 ( seconds )
#
# livedata_module_sla_provider__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, activez les options suivantes :

Nom	Type	Unité	Défaut	Commentaire
<code>livedata_module_sla_provider__use_ssh_tunnel</code>	Booléen	---	0	<ul style="list-style-type: none"> • 1 : Connexion par tunnel SSH • 0 : Connexion directe
<code>livedata_module_sla_provider__ssh_user</code>	Texte	Utilisateur unix	shinken	L'utilisateur avec lequel le tunnel sera établi
<code>livedata_module_sla_provider__ssh_keyfile</code>	Texte	Chemin de fichier	~shinken/.ssh/id_rsa	La clé SSH privée présente sur le serveur Shinken qui sera utilisé pour établir le tunnel.
<code>livedata_module_sla_provider__ssh_tunnel_timeout</code>	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 la reconnexion automatique 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 )
#
#   livedata_module_sla_provider__database__retry_connection_X_times_before_considering_an_error 4

#   Time between each try
#
#       Default : 3 ( seconds )
#
#   livedata_module_sla_provider__database__wait_X_seconds_before_reconnect 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
#   retry_connection_X_times_before_considering_an_error * wait_X_seconds_before_reconnect
#   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 déclenchée ce qui provoque une coupure temporaire de l'accès à la base.

Dans le but de ne pas interrompre le service, le module "livedata-module-sla-provider" va se reconnecter automatiquement au cluster MongoDB.

Pour ce faire il va faire un nombre d'essais égal au paramètre "livedata_module_sla_provider__database__retry_connection_X_times_before_considering_an_error" avec une pause de X secondes entre chaque essai (correspondant au paramètre "livedata_module_sla_provider__database__wait_X_seconds_before_reconnect"

).

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
livedata_module_sla_provider__database__retry_connection_X_times_before_considering_an_error	Entier	essais	4	Nombre d'essais de reconnexion à la base
livedata_module_sla_provider__database__wait_X_seconds_before_reconnect	Entier	secondes	3	Temps entre chaque essai 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.

Paramétrage des états Données manquantes (*Missing data*) et Shinken inactif (*Shinken inactive*)

```
# #
# SLA CALCULATION #
# #

# 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.
#
# livedata_module_sla_provider__no_data_period      include
```

Les états Données manquantes (*Missing data*) et Shinken inactif (*Shinken inactive*) ont été regroupés dans un paramètre. Ce paramètre correspond à la période durant laquelle Shinken n'a pas effectué les vérifications pour un check (plateforme Shinken éteinte, ou vérification du check désactivé grâce aux [Périodes de temps](#)). Le statut de ces checks est donc Données manquantes ("*Missing data*") ou Shinken inactif ("*Shinken inactive*").

Nom	Type	Unité	Défaut	Commentaire
<code>livedata_module_sla_provider__no_data_period</code>	Texte	---	include	<ul style="list-style-type: none">• 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 (de Shinken) n'impacte pas le taux de disponibilité• ok : On considère que même si la supervision ne l'a pas confirmé, le service est rendu