

Module broker--module-report-builder

Sommaire

Concept

Activation du module

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

Créer un nouveau module de type broker__module_report_builder

Configuration

Exemple de fichier de configuration

Détails des sections composant le fichier de configuration

Identification du module

Paramètres réseau

Autorisation dans le pare feu

Chiffrement des échanges réseau

Paramètre d'identification

Modules

Statistiques (pour la supervision)

Absorption des broks (information de supervision venant des Schedulers, options internes)

Les fichiers de log du Reactionner sont situés dans le dossier `/var/log/shinken/`. Pour plus d'informations, consultez la page [Fichiers Logs](#).

Démarrage

Lors du démarrage du démon, une ligne est disponible:

Démarrage du daemon

```
[2020-05-18 05:19:18] INFO : [daemon ] [START-DAEMON] The daemon (version=02.08.01-release.fr) is now
started as a daemon (detached from any shell) with pid=15412
[2020-05-18 05:19:18] INFO : [daemon-master] [ SYSTEM ] System resource number of open files is
set to (soft:1024 / hard:1024 ) (from parameter max_file_descriptor_limit)
[2020-05-18 05:19:18] INFO : [daemon-master] [ SYSTEM ] System resource number of process
/threads is set to (soft:unlimited / hard:unlimited ) (set at system max values)
```

Avec comme informations principales:

- Sa version
- Son numéro de PID
- Ses limites systèmes en nombre de fichiers/socket ouvrables, et le nombre max de processus/threads

Récupération des notifications et des événements

Pour récupérer les notifications et les événements à exécuter

- **Reactionner actif**
 - Le Reactionner va demander au Scheduler.
 - Le Reactionner indique un temps de travail disponible (en temps cpu).
 - Le Scheduler lui donne des notifications et événements (suivant le temps d'exécution moyen de ces actions constaté sur ce Reactionner). Il lui donne pour un temps inférieur ou égal au temps demandé.
 - Un log permet d'avoir ce nombre de notifications et le nombre d'événements récupérés. Ce log s'affiche même si aucun événement ou notification n'a été récupéré :

```
[2020-05-18 05:19:18] INFO : [daemon-master] [ ACTIONS ] [ scheduler-master ] [ GET ] Requesting
actions todo from this scheduler for 2.000s cpu time [received=11 notification(s) / 0 event(s) for 1.087s
cpu time]
```

- **Reactionner passif**
 - Le Scheduler demande au Reactionner le temps CPU disponible
 - Le Scheduler lui envoie des notifications et événements à traiter pour le temps récupéré (suivant le temps d'exécution moyen de ces actions constaté sur ce Reactionner). Il lui donne pour un temps inférieur ou égal au temps demandé.
 - Si des notifications ou événements sont reçus, un log permet d'avoir le nombre de notifications et le nombre d'événements récupérés selon le temps de travail disponible sur le Reactionner :

```
[2020-05-18 05:19:18] INFO : [daemon-master] [ ACTIONS ] [ scheduler-master ] [ RECEIVED ] We did received actions todo from this scheduler for 1.000s cpu time [received=2 notification(s) / 0 event(s) for 0.160s cpu time]
```

Envoi des résultats de notifications et d'événements au Scheduler

- **Reagir actif**
 - Une fois les notifications et événements exécutés, le Reagir envoie les résultats au Scheduler.
 - Un log permet d'avoir le nombre de résultats de notifications et d'événements envoyés au Scheduler et le temps mis pour être envoyé.

```
[2020-05-18 05:19:18] INFO : [daemon-master] [ ACTIONS RESULTS ] [scheduler-master] [ PUSHED ] 1 action's result(s) send to this scheduler in [0.043]s [1 notifications / 0 events]
```

- **Reagir passif**
 - Une fois les notifications et événements exécutés, le Reagir stock les résultats en attendant que le Scheduler vienne les récupérer.
 - A chaque tour de boucle du Scheduler, ce dernier demande au Reagir s'il a des résultats de notifications et événements disponibles.
 - Si des résultats sont disponibles, un log permet d'avoir le nombre de résultats de notifications et d'événements donné au Scheduler.

```
[2020-05-18 05:19:18] INFO : [daemon-master] [ ACTIONS RESULTS ] [scheduler-master] [ GIVEN ] 1 action's result(s) given to answer scheduler request [1 notifications / 0 events]
```

Surcharge serveur en activité disque, ralentissant l'écriture des logs

Si le serveur hébergeant le daemon est surchargé en termes d'IO disques sur le volume qui héberge le fichier de log, alors ce dernier va mettre du temps à s'écrire et va ralentir tout le daemon. Il faut alors si c'est faisable isoler le volume des disques sur un disque moins chargé pour ne pas ralentir le daemon.

En cas de soucis, vous aurez dans les logs l'entrée suivante:

```
2020-05-04 00:00:51 WARNING : [ LOGGER ]
2020-05-04 00:00:51 WARNING : [ LOGGER ]
-----
2020-05-04 00:00:51 WARNING : [ LOGGER ] [ WRITING ] The log write time is very high (1.87s). Please look at your log disk performance.
2020-05-04 00:00:51 WARNING : [ LOGGER ]
-----
2020-05-04 00:00:51 WARNING : [ LOGGER ]
```

Logs concernant les checks de vérifications Shinken

Quand un check de supervision du daemon est fait, on va avoir plusieurs entrées dans les logs qui concernent des données que le daemon garde sur diverses statistiques.

Un log permet d'avoir le temps pris sur le calcul des dernières commandes en timeout:

```
[2020-05-18 05:19:18] DEBUG : [daemon-master] [ STATS ] Compute "Checks in timeouts" stats : 0.000s in a total of 2048 commands in timeouts
```

Un log permet d'avoir le temps de calcul concernant les ranges d'exécution des checks/notifications en fonction du temps (<100ms, <400ms, etc):

```
[2020-05-18 05:19:18] DEBUG : [daemon-master] [ STATS          ] Compute "Checks per CPU running time" : 0.000s (on a total of 2048 checks)
```

Un log permet d'avoir le temps de calcul pour avoir les 5 commandes les plus longues en temps CPU:

```
[2020-05-18 05:19:18] DEBUG : [daemon-master] [ STATS          ] top5 execution time 0.003s (loop over 1 ranges and 343 elements)
```

Un dernier log permet d'avoir le temps complet du calcul des statistiques du daemon:

- Sur une partie commune à tous les daemons (version, chemins, etc)
- Sur la partie qui concerne uniquement ce qui concerne ce daemon, donc sont inclus ici les temps précédents
- Il est affiché en :
 - **DEBUG**: si le temps de calcul est inférieur à la valeur du paramètre **display_statistics_compute_time_if_higher** du daemon.
 - **INFO**: si le temps de calcul est supérieur ou égal au paramètre **display_statistics_compute_time_if_higher** du daemon.

```
[2020-05-18 05:19:18] DEBUG : [daemon-master] [ STATS          ] Daemon stats was computed in 0.020s (0.001 for daemon common part, 0.020 for reactionner part)
```

En cas d'affichage INFO on met un petit morceau en plus sur comment gérer le niveau de log:

```
[2020-05-18 09:26:27] INFO : [daemon-master] [ STATS          ] Daemon stats was computed in 0.004s (0.000 for daemon common part, 0.004 for poller part) (NOTE: log is display in INFO because 0.004 is higher than display_statistics_compute_time_if_higher=1ms in the daemon cfg)
```

Le daemon nettoie ses structures de statistiques toutes les 5minutes, ce qui sera vu par la ligne de log suivante:

```
[2020-07-01 16:47:16] INFO : [daemon-master] [ STATS          ] Clean checks in timeouts structure in 0.000s (before clean: 0 commands in timeouts, after clean: 0)
```

Logs de chargement des modules

Les démons ont une phase de chargement des modules qui est décrite dans la page [HIDDEN - Logs de chargement des modules](#)