

Voir la configuration du module (synchronizer-collector-vmware)

Échanges par packet des Broks avec le Broker (limite = taille maximum définie)

Sommaire

Concept

Les différentes parties de l'onglet

1 - Configuration générale

2 - Options de mélange des sources

Options de mélange des sources - Précisions

Mode de mélange des sources

Détecter les éléments qui ne sont plus présents dans la source

Définir une Période de maintenance sur les éléments supprimés

3 - Choix de l'espace où seront placés les éléments importés

Mettre dans l'espace - Précisions

4 - Clés spécifiques à la source

5 - Clés de synchronisation

Propriétés par défaut utilisé pour la construction des clés de synchronisation

Besoin de confirmation

Erreurs de configuration

Démarrage du démon

Au démarrage le démon affiche plusieurs logs contenant ses informations dont :

- Ses limites système en nombre de fichiers ouvrables, et nombre de threads/processus

```
[YYYY-MM-DD HH:MM:SS] INFO      : [daemon-master] [ SYSTEM          ] System resource number of open files is
set to      (soft:1024      / hard:1024      ) (from parameter max_file_descriptor_limit)
[YYYY-MM-DD HH:MM:SS] INFO      : [daemon-master] [ SYSTEM          ] System resource number of process
/threads is set to (soft:unlimited / hard:unlimited ) (set at system max values)
```

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

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

Si le serveur hébergeant le démon 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 démon. Il faut alors si c'est faisable isoler le volume des disques sur un disque moins chargé pour ne pas ralentir le démon.

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

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ LOGGER ]
[YYYY-MM-DD HH:MM:SS] WARNING : [ LOGGER ]
-----
[YYYY-MM-DD HH:MM:SS] WARNING : [ LOGGER ] [ WRITING ] The log write time is very high (1.87s). Please look
at your log disk performance.
[YYYY-MM-DD HH:MM:SS] WARNING : [ LOGGER ]
-----
[YYYY-MM-DD HH:MM:SS] WARNING : [ LOGGER ]
```

Chargement des broks initiaux par un regenerator (créateur d'objets des modules de broker) et vérifier que c'est bien la même configuration charger entre les regenerators / scheduler / arbiter

Les logs suivants permettent de suivre le chargement de la configuration de supervision entre l'Arbiter les Schedulers jusqu'aux interfaces : webui / livestatus / livedata

Il existe 2 types d'identifiants de configuration (représentation de la configuration)

- **configuration_uuid**: uuid de configuration total généré par l'Arbiter
- **configuration_part_id**: id de la partie de configuration géré par un Scheduler

Quand un Scheduler reçoit une nouvelle configuration de l'arbitre, il log

```
[YYYY-MM-DD HH:MM:SS] INFO : [scheduler] [schedulerdaemon] [ CONFIGURATION ] [ NEW ] Configuration received [ configuration_part_id=configuration_part_id, configuration_uid=configuration_uid, arbiter=arbiter_name, date=creation_date, ]
```

- **configuration_part_id**: id de la parti de configuration spécifiquement gérée par ce Scheduler (unique pour chaque Scheduler)
- **configuration_uid**: uuid créé lors du démarrage de l'Arbiter qui correspond donc à l'id de la configuration géré par l'Arbiter
- **creation_date**: date du démarrage de l'Arbiter
- **arbiter_name**: nom de l'Arbiter qui a créé cette configuration

Exemple Scheduler réception d'une nouvelle configuration

```
[YYYY-MM-DD HH:MM:SS] INFO : [scheduler] [schedulerdaemon] [ CONFIGURATION ] [ NEW ] Configuration received [ configuration_part_id=1280, configuration_uid=e551f7f93f2d45bfafae77fc302db7a2, arbiter=arbiter-master1, date=15-05-2020 15:13:38 ]
```

Quand un Scheduler a fini de charger la nouvelle configuration reçu, il log :

```
[YYYY-MM-DD HH:MM:SS] INFO : [scheduler-master1] [schedulerdaemon] [ CONFIGURATION ] [ NEW ] Loaded in [loading_time]s => [ configuration_part_id=configuration_part_id, configuration_uid=configuration_uid, date=creation_date, author=arbiter_name ]
```

- **configuration_part_id**: id de la parti de configuration géré par ce Scheduler (unique pour chaque Scheduler)
- **configuration_uid**: uuid crée lors du démarrage de l'Arbiter qui correspond donc à l'id de la configuration géré par l'Arbiter
- **creation_date**: date du démarrage de l'Arbiter
- **arbiter_name**: nom de l'Arbiter qui a créé cette configuration
- **loading_time**: temps de chargement de la configuration

Exemple Scheduler chargement de la nouvelle configuration

```
[YYYY-MM-DD HH:MM:SS] INFO : [scheduler-master1] [schedulerdaemon] [ CONFIGURATION ] [ NEW ] Loaded in [1.31168293953]s => [ configuration_part_id=1280 configuration_uid=e551f7f93f2d45bfafae77fc302db7a2, author=arbiter-master1, date=15-05-2020 15:13:38]
```

Logs DEBUG des escalades

Si le Scheduler est démarré avec la variable d'environnement

```
SHINKEN_SUPPORT_ENABLE_ESCALATION_DEBUG=1 /etc/init.d/shinken-scheduler start
```

Alors il va afficher plus d'informations DEBUG sur les escalades afin de pouvoir mieux les suivre.

Choix de l'escalade

Savoir quelle notification est prise, on a ce log. Attention, seule la dernière ligne sera utile (une escalade ayant pris le pas sur la précédente):

```
[YYYY-MM-DD HH:MM:SS0] DEBUG : ESCALATION: [NOM HOTE/CHECK] The escalation NOM_ESCALADE give us a valid start time (TEMPS). Looking if other escalations are giving us earlier time.'
```

Choix de l'intervalle

Savoir quel intervalle de notification est pris, il faut là encore prendre la dernière ligne (si plusieurs escalades sont possibles au même moment):

```
[YYYY-MM-DD HH:MM:SS] DEBUG : ESCALATION: [NOM HOTE/CHECK] The escalation NOM_ESCALADE is eligible and with a short notification interval, so we are using its interval: VALEUR_INTERVAL_ESCALADE.
```

Escalade avec un intervalle de notification à 0

Quand une escalade n'a pas de notification interval (ceci va envoyer une seule notification) on est mis au courant par cette ligne:

```
[YYYY-MM-DD HH:MM:SS] DEBUG : ESCALATION: [NOM HOTE/CHECK] The escalation NOM_ESCALADE have no notification interval, so disabling next notification.
```

Erreur de cohérence des périodes de maintenance

Les périodes de maintenance ont une incohérence entre non identifiée qui consiste à ce qu'une période de maintenance va être démarrée deux fois, ce qui va poser soucis lors de son arrêt (l'hôte/check sera dans un état incohérent). Pour l'instant, nous n'avons pas trouvé la source de l'incohérence, mais nous avons mis des protections pour les éviter et corriger.

Au chargement de la rétention, on aura une entrée "ERROR":

```
[YYYY-MM-DD HH:MM:SS] ERROR : [scheduler-master] [ DOWNTIME-INCOHERENCY ] The host moi has bad downtime values (saved number of downtime=1, actual=0). We are fixing the values. Please report it to the support.
```

Lors du second démarrage de la période de maintenance:

```
[YYYY-MM-DD HH:MM:SS] ERROR : [scheduler-master] [ DOWNTIME-INCOHERENCY ] The downtime [Downtime id=1596532528600972 active=active type=fixed start=Tue Aug 4 11:16:06 2020 - end=Tue Aug 4 12:15:06 2020] on moi got activated twice. This is a bug and MUST be reported to support for investigation. Thanks.
```

Dans les deux cas, il faut récupérer les logs du Scheduler, et les donner en analyse au support Shinken.

Exécution de commandes externes reçues d'un Receiver

Lorsque le Receiver envoie des commandes externes au Scheduler, il est possible que ce dernier n'ai pas reçu de configuration de la part de l'Arbiter et qu'il ne soit pas en capacité d'exécuter ces commandes.

```
[YYYY-MM-DD HH:MM:SS] INFO : [scheduler] [ EXTERNAL COMMANDS ] Get external commands from the Receiver receiver-master but i am not ready. Waiting for configuration from Arbiter.
```

Dans le cas où, les commandes ont pu être exécutées :

```
[YYYY-MM-DD HH:MM:SS] INFO : [scheduler-master] [ EXTERNAL COMMANDS ] Running 200 external commands received from the Receiver receiver-master
```

Échanges par packet de taille limité des Broks avec le Broker

Quand le broker demande les broks au Scheduler, ce dernier va avoir dans ses logs:

```
[YYYY-MM-DD HH:MM:SS] INFO : [scheduler-master] [ GIVE BROKS ] [ broker-master ] Sending 5 broks (1.3kB)
```

Si on se retrouve dans un cas où la limite du **broks_packet_size** est atteinte (sur les broks initiaux par exemple), on va avoir la ligne suivante:

```
[YYYY-MM-DD HH:MM:SS] INFO : [ scheduler4-master1 ] [ GIVE BROKS ] [ broker-master3 ] Sending 2588 broks (413.4kB) [chunk, still 1698 to send]
```

En cas de problème de communication, il se peut qu'un paquet de broks soit perdu, dans ce cas, une réémission est faite pour éviter cette perte, on aura un WARNING suivant:

```
[YYYY-MM-DD HH:MM:SS] WARNING: [scheduler-master] [ GIVE BROKS ] [ broker-master ] Packet number did mismatch "100d0ccf12de4665bf04f2150dcc97d5" != "0ca27bc3ea5440358c1194b5b7c3b4f4" : Re-sending broks (6.3kB)
```

Les logs de la retention mongo

Connexion à une base de donnée

Quand le module mongo se connecte à une base de données, on va avoir le log suivant:

```
[YYYY-MM-DD HH:MM:SS] INFO : [scheduler-master] [ MongodBRetention ] [ SAVE WORKER 1 ] We are creating mongo connection [uri=mongodb://127.0.0.1/?safe=false] [database=shinken_retention] [ssh=False]
```

Il y indique donc:

- L'URL utilisée
- La base de données (peut être différente du défaut "shinken" comme ici)
- Si un tunnel SSH va être utilisé ou pas

Log de performance de la boucle du scheduler

Afin d'avoir toutes les informations de DEBUG sur les performances du scheduler (boucle principale) il faut le lancer avec:

```
SHINKEN_LOG_SCHEDULER_RECURRENT_TIMES_FLAG=1 /etc/init.d/shinken-scheduler -d start
```