

Receiver - \$KEY\$ - Alive

Sommaire

- Contexte
- Paramétrage
 - Données utilisées provenant du modèle
 - Données communes pour les checks du modèle
 - Provenant du modèle shinken
 - Provenant du modèle shinken-daemon
 - Provenant du modèle shinken-receiver
 - Données spécifiques pour ce check
 - Les données DFE (Duplicate Foreach)
 - Données utilisées provenant du check
 - Données globales
 - Propriétés de l'hôte
- Résultat
 - Exemple
 - Interprétation
 - Statut
 - Les vérifications communes
 - Les vérifications spécifiques
 - Résultat
 - Résultat Long
 - Description des erreurs
 - Erreur d'un démon bloqué, qui doit être redémarré
 - Un démon spare n'est assigné sur aucun démon master
 - Problème de conflits d'Arbiters
 - Les serveurs ne sont pas à la même heure
 - Erreur de surcharge des disques de logs
 - Le démon a bloqué une tentative de chargement d'objet malveillant
 - Le démon est en cours d'arrêt
 - La dernière connexion de l'Arbiter remonte à trop longtemps
 - Le démon a chargé une configuration enregistrée
 - Erreur de vol de CPU

- Métriques

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 terme 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 lots 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 ]
```

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 permet de suivre le chargement de la configuration de supervision entre l'Arbiter les schedulers jusqu'au interface : webui / livestatus / livedata

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

- **configuration_uid**: 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

```
[1589548527] INFO : [scheduler] [schedulerdaemon] [ CONFIGURATION ] [ NEW ] Configuration received [
configuration_part_id=configuration_part_id, configuration_uuid=configuration_uuid,
arbiter=arbiter_name, date=creation_date, ]
```

- **configuration_part_id**: id de la parti de configuration spécifiquement gérée par ce Scheduler (unique par Scheduler)
- **configuration_uuid**: 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

Exemple Scheduler réception d'une nouvelle configuration

```
[1589548527] INFO : [scheduler] [schedulerdaemon] [ CONFIGURATION ] [ NEW ] Configuration received [
configuration_part_id=1280, configuration_uuid=e551f7f93f2d45bfafae77fc302db7a2, arbiter=arbiter-master1,
date=15-05-2020 15:13:38 ]
```

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

```
[2020-05-15 15:15:30] INFO : [scheduler-master1] [schedulerdaemon] [ CONFIGURATION ] [ NEW ] Loaded in
[loading_time]s => [ configuration_part_id=configuration_part_id, configuration_uuid=configuration_uuid,
date=creation_date, author=arbiter_name ]
```

- **configuration_part_id**: id de la partit de configuration géré par ce Scheduler (unique par Scheduler)
- **configuration_uuid**: 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

```
[2020-05-15 15:15:30] INFO : [scheduler-master1] [schedulerdaemon] [ CONFIGURATION ] [ NEW ] Loaded in
[1.31168293953]s => [ configuration_part_id=1280 configuration_uuid=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):

```
[2020-05-15 15:15:30] 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 quelle intervalle de notification est pris, il faut là encore prendre la dernière ligne (si plusieurs escalades sont possibles au même moment):

```
[2020-05-15 15:15:30] 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:

```
[2020-05-15 15:15:30] DEBUG : ESCALATION: [NOM HOTE/CHECK] The escalation NOM_ESCALADE have no notification interval, so disabling next notification.
```