

# shinken-broker ( Modèle d'hôte )

## Sommaire

- Contexte
- Description du modèle
  - Checks
    - Un démon spare n'a pas à avoir la même liste de module que son master
  - Données du modèle
  - Métriques enregistrées
  - Commandes
- Check : Broker - \$KEY\$ - Alive
  - Description des erreurs
    - Erreur de surcharge des disques de logs
    - Erreur de vol de CPU
    - Un démon spare n'est assigné sur aucun démon master
    - Problème de conflits d'Arbiters
    - Les serveurs ne sont pas à la même heure
    - La dernière connexion de l'Arbiter remonte à trop longtemps
    - Erreur d'un démon bloqué, qui doit être redémarré
    - Le démon a bloqué une tentative de chargement d'objet malveillant
    - Le démon est en cours d'arrêt

## Contexte

Plusieurs modèles vous permettent de superviser un hôte hébergeant le démon Broker ( voir la page [Le Broker](#) ).

- Le modèle **shinken-broker** pour vérifier le comportement du démon..
- Le modèle **shinken-broker-module-metrology-writer** attachera un check additionnel qui permettra de superviser l'écriture des données de métrologie.
- Le modèle **shinken-broker-module-sla-writer** attachera un check additionnel qui permettra de superviser l'écriture des données de SLA.
- Le modèle **shinken-broker-module-visualisation-ui** attachera un check additionnel qui permettra de superviser la lecture des données de métrologie.
  - Le modèle **shinken-broker-module-visualisation-ui-sla-reader** attachera un check additionnel qui permettra de superviser la lecture des données SLA.
  - Le modèle **shinken-broker-module-visualisation-ui-event-manager-reader** attachera un check additionnel qui permettra de superviser la lecture des données du bac à événement.
- Le modèle **shinken-broker-module-event-manager-writer** attachera un check additionnel qui permettra de superviser l'écriture des données du bac à événement.
- Le modèle **shinken-broker-db** attachera quant à lui, les checks qui permettront de superviser la base utilisée par Shinken pour le démon Broker.

## Description du modèle

Modèle d'hôte correspondant: **shinken-broker** (notez que ce modèle hérite du modèle **shinken**, **shinken-deamon** et **shinken-broker-db** )

Afin de superviser le démon Broker, le modèle **shinken-broker** appliqué à votre hôte, attachera plusieurs checks qui vérifieront la santé et la performance de ce démon.

## Checks

### Broker - \$KEY\$ - Alive

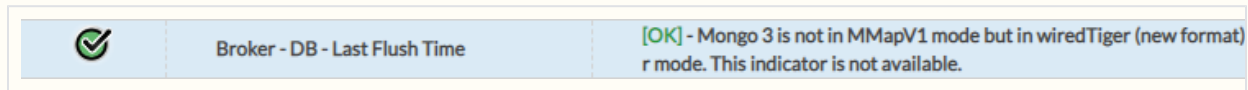
Vérifie que le démon Broker peut être correctement contacté sur le réseau, que la version du démon est affichée dans le ( *Résultat court* ) et aussi que les modules sont opérationnels ( *Résultat long* ).





La version de Shinken est accompagnée d'une mise à jour de MongoDB. Cette nouvelle version comporte notamment un changement de la méthode de sauvegarde des objets au sein de Mongo.

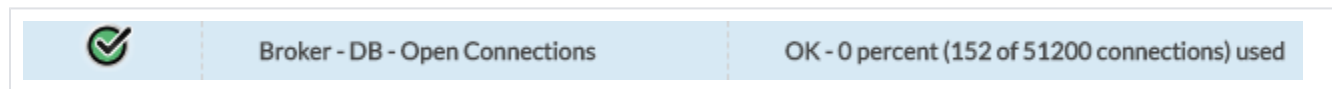
Sur une installation neuve de Shinken, le nouveau format (WiredTiger) est utilisé, ce qui rend le "*Broker - DB - Last Flush Time*" non pertinent. Dans ce cas, ce check aura le résultat suivant :



Dans le cas d'une mise à jour de Shinken, la méthode de stockage des données dans Mongo n'est pas modifiée et le check "*Broker - DB - Last Flush Time*" a le même comportement que précédemment.

### Broker - DB - Open Connections

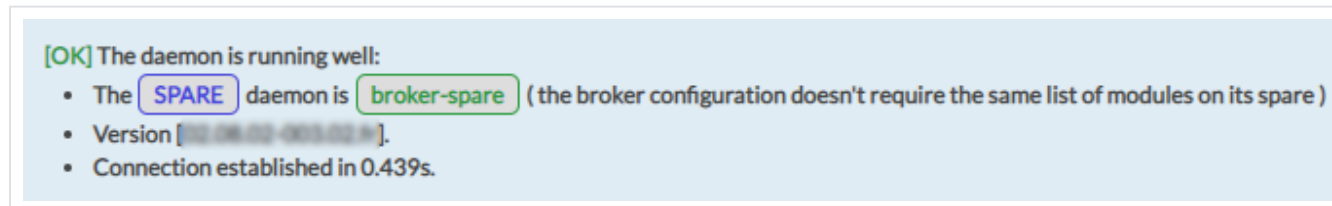
Vérifie le nombre de connexions en cours sur la base de données.



Lors de la mise à jour de Shinken une montée de version de MongoDB est effectuée. Le check "Broker - DB - Lock Time Percentage" devient obsolète et peut être désactivé. La source "cfg-file-shinken" propose la désactivation du check, mais ne peut pas le supprimer automatiquement.

### Un démon spare n'a pas à avoir la même liste de module que son master

Quand le master est configuré pour avoir un spare qui n'a pas la même liste de modules que lui, le check ajoute un message d'information le précisant :



Consulter l'option suivante dans la configuration du master ( voir la page [Le Broker](#) ).

```
broker__manage_spare__spare_must_have_the_same_list_of_module_type
```

### Données du modèle

Les checks du Broker peuvent être configurés via des données fournies par le modèle.

Les données suivantes sont disponibles :

Nom de la donnée	Description	Valeur par défaut	Hérité du modèle d'hôte ou locale
SHINKEN_PROTOCOL	Protocole utilisé pour établir la connexion avec le Broker	http	shinken
CHECK_SHINKEN_TIMEOUT	Timeout utilisé pour l'établissement de la connexion avec Broker	3	shinken
BROKER_PORT	Port utilisé pour l'établissement de la connexion avec Broker	7772	Locale
BROKER_LIST	Liste de Broker ( <i>Multi-démon</i> )	broker-master\$( \$_HOSTBROKER_PORT )\$	Locale - Dupliquer des checks en fonction d'une liste de valeurs présentes dans la Donnée d'un hôte (duplicate_foreach)
DB_CONNECTION_METHOD	Méthode de connexion au serveur MongoDB ( " <i>direct</i> " ou " <i>ssh</i> " )	ssh	shinken-broker-db
DB_CONNECTION_S_CRIT	Seuil critique du nombre de connexions en cours sur la base de données ( <i>en pourcentage</i> )	80	shinken-broker-db

DB_CONNECTION_S_WARN	Seuil d'avertissement du nombre de connexions en cours sur la base de données ( <i>en pourcentage</i> )	70	shinken-broker-db
DB_CONNECT_CRIT	Seuil critique pour le temps de connexion à la base de données ( <i>en secondes</i> )	4	shinken-broker-db
DB_CONNECT_WARN	Seuil d'avertissement pour le temps de connexion à la base de données ( <i>en secondes</i> )	2	shinken-broker-db
DB_LAST_FLUSH_TIME_CRIT	Seuil critique pour le temps de flush de la base de données ( <i>en millisecondes</i> )	10000	shinken-broker-db
DB_LAST_FLUSH_TIME_WARN	Seuil d'avertissement pour le temps de flush de la base de données ( <i>en millisecondes</i> )	8000	shinken-broker-db
DB_PORT	Port de la base de données	27017	shinken-broker-db
DB_SSH_KEY	Clé SSH privée permettant la connexion au serveur hébergeant MongoDB	~/ssh/id_rsa	shinken-broker-db
DB_SSH_USER	Utilisateur distant avec lequel se connecter au serveur hébergeant MongoDB	shinken	shinken-broker-db
THRESHOLD_CPU_STOLEN_WARNING	Seuil de cpu volé ( <i>en pourcentage</i> ) sur une machine virtuelle supervisée par vmware avant de déclencher un warning	5	shinken-deamon
THRESHOLD_CPU_STOLEN_CRITICAL	Seuil de cpu volé ( <i>en pourcentage</i> ) sur une machine virtuelle supervisée par vmware avant de déclencher un critique	10	shinken-deamon

Voir la page de [Sécurisation des connexions aux bases MongoDB](#) pour plus d'information sur un paramétrage sécurisé.

## Métriques enregistrées

Les checks du modèle enregistrent des données de performance, qui peuvent ensuite être affichées dans l'interface de Visualisation sur l'[Onglet Graphiques](#) ou bien le [Widget Graphique](#).

Nom du check	Nom de la métrique	Explication
Broker - \$KEY\$ - Alive	connexion_time	Temps de connexion en secondes pour contacter le démon
Broker - \$KEY\$ - Alive	cpu_stolen_vmware_percent_ready	Seulement si le démon est situé sur une VM VMWare Valeur de l'indicateur VMWare %ready ( <i>temps de blocage de la VM avant d'avoir accès à ses VCpu, donc temps perdu du point de vue de la VM</i> )
Broker - \$KEY\$ - Performance API Connection	get_lock_time	Temps de connexion et d'obtention d'un appel bloquant dans le démon et ainsi voir si les appels bloquants ne sont pas trop long
Broker - \$KEY\$ - Performance Modules Queues	*.data_queue	Taille de la file d'attente des envois de Broks vers les modules externes. Un élément étant rajouté par seconde, ceci représente le nombre de secondes de Brok reçus qui n'ont pas encore été récupérés par le module.

## Commandes

Nom du check	Commande du check	Ligne de commande
Broker - \$KEY\$ - Alive	check_shinken_broker! <b>alive!</b> <b>\$VALUE1\$</b>	\$PLUGINSDIR\$/check_shinken -H "\$HOSTADDRESS\$" -p "\$ARG2\$" --shinkenversion "\$SHINKENVERSION\$" -t broker -m \$ARG1\$ --timeout \$_HOSTCHECK_SHINKEN_TIMEOUT\$ -w \$_HOSTTHRESHOLD_CPU_STOLEN_WARNING\$ -c \$_HOSTTHRESHOLD_CPU_STOLEN_CRITICAL\$
Broker - \$KEY\$ - Performance API Connection	check_shinken_broker! <b>api_connection!</b> <b>\$VALUE1\$</b>	\$PLUGINSDIR\$/check_shinken -H "\$HOSTADDRESS\$" -p "\$ARG2\$" --shinkenversion "\$SHINKENVERSION\$" -t broker -m \$ARG1\$ --timeout \$_HOSTCHECK_SHINKEN_TIMEOUT\$ -w \$_HOSTTHRESHOLD_CPU_STOLEN_WARNING\$ -c \$_HOSTTHRESHOLD_CPU_STOLEN_CRITICAL\$
Broker - \$KEY\$ - Performance Modules Queues	check_shinken_broker! <b>modules_queue!</b> <b>\$VALUE1\$</b>	\$PLUGINSDIR\$/check_shinken -H "\$HOSTADDRESS\$" -p "\$ARG2\$" --shinkenversion "\$SHINKENVERSION\$" -t broker -m \$ARG1\$ --timeout \$_HOSTCHECK_SHINKEN_TIMEOUT\$ -w \$_HOSTTHRESHOLD_CPU_STOLEN_WARNING\$ -c \$_HOSTTHRESHOLD_CPU_STOLEN_CRITICAL\$

Broker - DB - Connection	check_shinken db_connection	\$PLUGINS_DIR\$/check_shinkendb.py -D -H "\$HOSTADDRESS\$" -P "\$_HOSTDB_PORT\$" --connection-method "\$_HOSTDB_CONNECTION_METHOD\$" --ssh-user "\$_HOSTDB_SSH_USER\$" --ssh-keyfile "\$_HOSTDB_SSH_KEY\$" -A connect -W "\$_HOSTDB_CONNECT_WARN\$" -C "\$_HOSTDB_CONNECT_CRIT\$" --timeout \$_HOSTCHECK_SHINKEN_TIMEOUT\$
Broker - DB - Last Flush Time	check_shinken db_last_flush	\$PLUGINS_DIR\$/check_shinkendb.py -D -H "\$HOSTADDRESS\$" -P "\$_HOSTDB_PORT\$" --connection-method "\$_HOSTDB_CONNECTION_METHOD\$" --ssh-user "\$_HOSTDB_SSH_USER\$" --ssh-keyfile "\$_HOSTDB_SSH_KEY\$" -A last_flush_time -W "\$_HOSTDB_LAST_FLUSH_TIME_WARN\$" -C "\$_HOSTDB_LAST_FLUSH_TIME_CRIT\$" --timeout \$_HOSTCHECK_SHINKEN_TIMEOUT\$
Broker - DB - Open Connections	check_shinken db_open_conn ections	\$PLUGINS_DIR\$/check_shinkendb.py -D -H "\$HOSTADDRESS\$" -P "\$_HOSTDB_PORT\$" --connection-method "\$_HOSTDB_CONNECTION_METHOD\$" --ssh-user "\$_HOSTDB_SSH_USER\$" --ssh-keyfile "\$_HOSTDB_SSH_KEY\$" -A connections -W "\$_HOSTDB_CONNECTIONS_WARN\$" -C "\$_HOSTDB_CONNECTIONS_CRIT\$" --timeout \$_HOSTCHECK_SHINKEN_TIMEOUT\$

Check : Broker - \$KEY\$ - Alive

## Description des erreurs

### Erreur de surcharge des disques de logs

En cas de disques trop lent sur le volume des logs, le check sera mis en **WARNING** avec l'erreur suivante.

**[WARNING]** The daemon have some issues.

=> **WARNING** - Writing logs on disk took too much time ( worth time was 2.0s during the last minute)  
Path: "/var/log/shinken/"

### Erreur de vol de CPU

Seulement si votre machine virtuelle est hébergé sur un hyperviseur VMWare

- **Votre machine à du vol de CPU :**  
Lorsque le CPU se fait voler trop de temps de calcul, le check sera mis en **WARNING** ou en **CRITIQUE** (*en fonction du taux de vol*) avec l'erreur suivante.
  - Vous pouvez avoir plus d'information sur cet indicateur et comment réduire la part de temps de la VM sur la page [Machine VMWare avec un fort taux de CPU Stolen \(%ready + %costop\)](#)

**[WARNING]** The daemon have some issues:

=> Your machine got **8% of CPU STOLEN** from the Hypervisor ( *Type VMWare* )  
→ On the VCenter search the data **CPU %ready + %costop**  
→ Please have a look at the Shinken Enterprise documentation about advices to reduce it

**[CRITICAL]** The daemon have some issues:

=> Your machine got **20% of CPU STOLEN** from the Hypervisor ( *Type VMWare* )  
→ On the VCenter search the data **CPU %ready + %costop**  
→ Please have a look at the Shinken Enterprise documentation about advices to reduce it

### Un démon spare n'est assigné sur aucun démon master

Un spare n'est pas assigné à un démon master:

Lorsqu'un démon Broker spare n'est assigné par aucun démon master, il est inutile, et remonte en erreur dans le check, tout comme dans la commande shinken-healthcheck, afin qu'il soit rapidement identifié et corrigé.

**SPARE** **[CRITICAL]**

No master is using this spare daemon → **UNUSED**

## Problème de conflits d'Arbiters

- **Conflits d'Arbiters :**

Si le démon est contacté par des Arbiters qui ne sont pas sur la même architecture ( *par exemple un Arbiter de Production et un autre de l'environnement de Testing* ), le check sera mis en **CRITICAL** .

```
=> Arbiters CONFLICT
Architecture List :
1. Production [194.168.1.17]
  o arbiter-master : last connection 11s ago. Defined on the server with uuid d2a358b0-cae3-4ce8-bc3f-edd241627bac (/var/lib/shinken/server.uuid)
2. Testing [194.168.1.17]
  o arbiter-master : last connection 12s ago. Defined on the server with uuid 785e0227-455a-449d-b8f1-516897000e3b (/var/lib/shinken/server.uuid)
```

- **Conflit d'Arbiters qui ont le même nom d'Architecture :**

Comme dans le cas précédent, le démon est contacté par des Arbiters d'architectures différentes, mais qui ont le même nom. On sort également en CRITICAL mais en avertissant que les noms sont identiques, et en indiquant où changer le nom de vos architectures.

```
=> Arbiters CONFLICT
Architecture List :
1. Production [194.168.1.17]
  o arbiter-master : last connection 33s ago. Defined on the server with uuid d2a358b0-cae3-4ce8-bc3f-edd241627bac (/var/lib/shinken/server.uuid)
2. Production [194.168.1.17]
  o arbiter-master : last connection 10s ago. Defined on the server with uuid 785e0227-455a-449d-b8f1-516897000e3b (/var/lib/shinken/server.uuid)

NOTE:
Some architecture have the same name. We advise you to change it in the configuration of their architecture_export module.
```

## Les serveurs ne sont pas à la même heure

Si le serveur n'est pas à la même heure que le serveur Arbiter ( *qui fait office de référence* ), une erreur **CRITICAL** sera levée, car des temps différents sur les différents serveurs va avoir des effets **désastreux** sur la cohérence des données de supervision.

```
=> Arbiters connection :
• Architecture Production :
  o ERROR arbiter-master => server times are different, time shift of 1 days 16h
```

## La dernière connexion de l'Arbiter remonte à trop longtemps

Si la dernière connexion de l'Arbiter remonte à trop de temps, le démon va lever un **WARNING** . Ceci peut être dû :

- Les Arbiters MASTER et SPARE sont réellement éteints
- Les Arbiters MASTER et SPARE sont en train d'envoyer des configurations à d'autres démons, et ne peuvent donc pas contacter ce démon pour l'instant.

=> Arbiters connection :

- Architecture *Production* :
  - arbiter-master => Missed connection from arbiter since 1 days 6h ( > daemon check\_interval \* max\_check\_attempts )



Le temps pris en compte comme limite de dernière connexion est de `check_interval * max_check_attempts` du démon ( définis dans sa configuration ).

Les valeurs par défauts sont de `60s * 3`, soit 3 minutes.

### Erreur d'un démon bloqué, qui doit être redémarré

- Si un démon est dans un état bloqué, il doit être redémarré. Si c'est le cas:
  - les checks seront en **ERROR** avec le message suivant,
  - il faut ouvrir un ticket à votre support pour analyser le blocage

**[CRITICAL]**

The daemon have a **lock**, it's **not working** and **MUST** be restarted.

Please contact your support to analyse the daemon logs:

- "Main loop" was locked more than 3600s
- Detected at 2021-12-03 08:21:55 [ WATCH DOG ]

### Le démon a bloqué une tentative de chargement d'objet malveillant

Il est possible qu'un démon puisse détecter et bloquer une tentative d'injection d'objet malveillant par le biais de l'une de ses routes.

Un message est remonté :

- le nombre total de ces tentatives que le démon a bloqué ce jour ( le compte commence à minuit ) ;
- pour chacune des tentatives ( *maximum 3* ) :
  - descriptif de l'objet que l'attaquant essaye de charger,
  - sa provenance de l'attaque, par exemple le nom de la route utilisée, et l'IP à la source de l'attaque,
  - sa date.

**[WARNING]** The daemon have some issues:

=> There were [ 1 ] security breaches blocked today (last 3):

- [ hacker/attack ] by [ HTTP(s) call "Configuration reception from an Arbiter server" by IP=127.0.0.1 ] at [ 2022-02-21 14:44:39 ]

### Le démon est en cours d'arrêt

Lorsque le démon est en cours d'arrêt, le check le signale, et les informations relatives aux modules ne sont plus disponibles

**[WARNING]** The broker is performing a shutdown.