

Détail d'un check

Sommaire

- Concept
- La barre d'information (partie haute)
- Onglet "Gérer"
 - Prise en compte
 - Période de Maintenance
- Onglet "Contacts et Notifications"
 - Informations sur la configuration du check
 - Utilisateurs et groupes à notifier
 - Période de notification
 - Informations sur les incidents et notifications
 - Incidents
 - Notifications
 - Détails des incidents et notifications
 - Incidents
 - Notifications isolées
 - Filtrer les destinataires
- Onglet "Avancé"
- Fermer la vue

Contexte

Plusieurs modèles vous permettent de superviser un hôte hébergeant le démon [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*).

Si jamais le démon Arbiter est en exécution sur une machine virtuelle supervisée par VMware, alors le pourcentage de temps de vol de CPU (*CPU % Ready*) sera affiché.

? Unknown Attachment

Broker - \$KEY\$ - Performance API Connection

Vérifie les temps de connexion vers le Broker.

? Unknown Attachment

Broker - \$KEY\$ - Performance Modules Queues

Vérifie les performances des modules du Broker.

? Unknown Attachment

Broker - DB - Connection

Vérifie le temps de connexion à la base de données.

? Unknown Attachment

Broker - DB - Last Flush Time

Vérifie la durée du dernier "Flush" de la base de données.

? Unknown Attachment



La version V02.07.00 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 :

? Unknown Attachment

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.

? Unknown Attachment



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 :

? Unknown Attachment

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 (Multi-démon)	broker-master\${\$_HOSTBROKER_PORT}\$	Locale - Duplicate For Each
DB_CONNECTION_METHOD	Méthode de connexion au serveur MongoDB ("direct" ou "ssh")	ssh	shinken-broker-db
DB_CONNECTIONS_CRIT	Seuil critique du nombre de connexions en cours sur la base de données (en pourcentage)	80	shinken-broker-db
DB_CONNECTIONS_WARN	Seuil d'avertissement du nombre de connexions en cours sur la base de données (en pourcentage)	70	shinken-broker-db
DB_CONNECT_CRIT	Seuil critique pour le temps de connexion à la base de données (en secondes)	4	shinken-broker-db
DB_CONNECT_WARN	Seuil d'avertissement pour le temps de connexion à la base de données (en secondes)	2	shinken-broker-db
DB_LAST_FLUSH_TIME_CRIT	Seuil critique pour le temps de flush de la base de données (en millisecondes)	10000	shinken-broker-db
DB_LAST_FLUSH_TIME_WARN	Seuil d'avertissement pour le temps de flush de la base de données (en millisecondes)	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 (en pourcentage) sur une machine virtuelle supervisée par vmware avant de déclencher un warning	5	shinken-deamon
THRESHOLD_CPU_STOLEN_CRITICAL	Seuil de cpu volé (en pourcentage) 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 Graphes](#) 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 (temps de blocage de la VM avant d'avoir accès à ses VCpu, donc temps perdu du point de vue de la VM)
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!alive! \$VALUE1\$	\$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!api_connection!\$V ALUE1\$	\$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!modules_queue!\$V ALUE1\$	\$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	\$PLUGINSDIR\$/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	\$PLUGINSDIR\$/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_connections	\$PLUGINSDIR\$/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\$

Description des erreurs de Broker - \$KEY\$ - Alive

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.

? Unknown Attachment

Erreur de vol de CPU

Seulement si votre machine virtuelle est hébergé sur un hyperviseur VMWare

Votre machine à du vol de CPU :

Si 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 fixé par défaut ou indiqué par l'utilisateur).

? Unknown Attachment

? Unknown Attachment

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 le shinken-healthcheck, afin qu'il soit rapidement identifié et corrigé.

? Unknown Attachment

Problème de conflits d'Arbiters

Conflit d'Arbiters :

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

? Unknown Attachment

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

Comme dans le cas précédent, le démon est contacté par des Arbiters d'architectures différents, mais qui ont le même nom. On sort également en **CRITICAL** mais en avertissant que les noms sont identiques, et en indiquant comment retrouver les serveurs en question, en trouvant leur valeur dans le fichier `/var/lib/shinken/server.uuid`.

? Unknown Attachment

Les serveurs ne sont pas à la même heure

Si le serveur n'est pas à la même heure que le serveur Arbitre (*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 cohérences des données de supervision.

? Unknown Attachment

La dernière connexion de l'Arbitre remonte à trop longtemps

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

- Les arbiters MASTER et SPARE sont réellement éteints
- Les arbitre 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.

? Unknown Attachment



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.