

# Synchronizer - \$KEY\$ - Performance API Connection

## 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-synchronizer

- 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

## Contexte

Le check **Synchronizer - \$KEY\$ - Performance API Connection** vérifie la latence de connexion au Synchronizer et ses performances

Statut	Nom de check	Résultat	Résultat Long
	Synchronizer - synchronizer-lab-validationsw1 - Performance API Connection	[OK] API Connection is in the good range (0.010 < 2s).	-

## Paramétrage

Le check utilise la ligne de commande suivante :

```
$PLUGINSDIR$/check_shinken -H "$HOSTADDRESS$" -p "$ARG2$" --shinkenversion "$SHINKENVERSION$" -t synchronizer -m $ARG1$ --timeout "$_HOSTCHECK_SHINKEN_TIMEOUT$" -w "$_HOSTTHRESHOLD_CPU_STOLEN_WARNINGS$" -c "$_HOSTTHRESHOLD_CPU_STOLEN_CRITICAL$"
```

## Données utilisées provenant du modèle

### Données communes pour les checks du modèle

Provenant du modèle **shinken**

Nom	Modifiable sur	Unité	Défaut	Valeur par défaut à l'installation de Shinken	Description
CHECK_SHINKEN_TIMEOUT	l'Hôte ( Onglet Données )	seconde	3	3	Temps maximum durant lequel le check va attendre une réponse du démon.

#### Provenant du modèle **shinken-daemon**

Nom	Modifiable sur	Unité	Défaut	Valeur par défaut à l'installation de Shinken	Description
THRESHOLD_CPU_STOLEN_CRITICAL	l'Hôte ( Onglet Données )	%	10	10	Seuil de CPU volé sur une machine virtuelle supervisée par VMware avant de déclencher un critique.
THRESHOLD_CPU_STOLEN_WARNING	l'Hôte ( Onglet Données )	%	5	5	Seuil de CPU volé sur une machine virtuelle supervisée par VMware avant de déclencher un warning.

#### Provenant du modèle **shinken-synchronizer**

Nom	Modifiable sur	Unité	Défaut	Valeur par défaut à l'installation de Shinken	Description
SYNCHRONIZER_PORT	shinken-synchronizer ( Onglet Données )	---	7772	7772	Port utilisé pour l'établissement de la connexion avec Synchronizer  Check(s) impacté(s) : <ul style="list-style-type: none"> <li>Synchronizer - \$KEY\$ - Alive</li> <li>Synchronizer - \$KEY\$ - Performance API Connection</li> </ul>

#### Données spécifiques pour ce check

Pas de données spécifiques pour ce check.

#### Les données DFE ( Duplicate Foreach )

Nom	Modifiable sur	Unité	Défaut	Valeur par défaut à l'installation de Shinken	Description
SYNCHRONIZER_LIST	l'Hôte ( Onglet Données )	---	synchronizer-master\$( \$_HOSTSYNCHRONIZER_PORT\$ )\$	synchronizer-master\$( \$_HOSTSYNCHRONIZER_PORT\$ )\$	Liste des Synchronizers ( Multi-démon )  Check(s) impacté(s) : <ul style="list-style-type: none"> <li>Synchronizer - \$KEY\$ - Alive</li> <li>Synchronizer - \$KEY\$ - Performance API Connection</li> </ul>

#### Données utilisées provenant du check

Pas de données provenant du check pour ce check.

## Données globales


Nom	Modifiable sur	Unité	Défaut	Valeur par défaut à l'installation de Shinken	Description
PLUGINS_DIR	Non modifiable ( Sauf Admin Shinken )	---	/var/lib/shinken/libexec	<b>/var/lib/shinken/libexec</b>	Chemin absolu du dossier contenant la sonde ( <i>non modifiable</i> ).
SHINKEN_VERSION	Non modifiable	---	---	---	Numéro de version du shinken utilisé pour la comparaison avec le shinken surveillé.

## Propriétés de l'hôte

Nom	Modifiable sur	Unité	Défaut	Valeur par défaut à l'installation de Shinken	Description
HOSTADDRESS	l'Hôte ( Onglet Général )	---	Nom de l'hôte	<b>Nom de l'hôte</b>	Adresse de l'hôte

## Résultat

### Exemple

Statut	Nom de check	Résultat	Résultat Long
	Synchronizer - synchronizer-lab-validationsw1 - Performance API Connection	<b>[OK]</b> API Connection is in the good range (0.010 < 2s).	-

## Interprétation

### Statut

Il peut prendre deux valeurs **OK** / **CRITIQUE** / **ATTENTION** / **INCONNU**.

- Le statut va dépendre du retour de sonde et de la configuration spécifique du check pour les données suivantes :
  - HOSTCHECK\_SHINKEN\_TIMEOUT
  - THRESHOLD\_CPU\_STOLEN\_CRITICAL
  - THRESHOLD\_CPU\_STOLEN\_WARNING
- Voici un tableau récapitulatif du statut attendu suivant le retour de sonde :

### Les vérifications communes

Situation	Statut
Démon bloqué et qui doit être redémarré	<b>CRITIQUE</b>
Démon spare non assigné sur un démon master	<b>CRITIQUE</b>
Problème de conflits d'Arbiters	<b>CRITIQUE</b>
Les serveurs ne sont pas à la même heure	<b>CRITIQUE</b>
Erreur de surcharge des disques de logs	<b>ATTENTION</b>
Démon qui bloque une tentative de chargement d'objet malveillant	<b>ATTENTION</b>

Démon est en cours d'arrêt	<b>ATTENTION</b>
Dernière connexion de l'Arbiter remonte à trop longtemps	<b>ATTENTION</b>
Le démon a chargé une configuration enregistrée, en attendant d'être contacté par l'Arbiter	<b>ATTENTION</b>

#### Les vérifications spécifiques

Situation	Statut
En fonction du pourcentage de CPU volé : <ul style="list-style-type: none"> <li>• Si c'est <b>supérieur</b> à THRESHOLD_CPU_STOLEN_CRITICAL( <i>par défaut : 10%</i> )</li> </ul>	<b>CRITIQUE</b>
En fonction du pourcentage de CPU volé : <ul style="list-style-type: none"> <li>• Si c'est <b>supérieur</b> à THRESHOLD_CPU_STOLEN_WARNING( <i>par défaut : 5%</i> )</li> </ul>	<b>ATTENTION</b>
Si la sonde n'a pas eu de réponse avant le temps maximum <ul style="list-style-type: none"> <li>• Si <b>supérieur</b> à HOSTCHECK_SHINKEN_TIMEOUT ( <i>par défaut : 3 sec</i> )</li> </ul>	<b>INCONNU</b>

#### Résultat

Renvoi au format texte :

- Le temps de réponse du Synchronizer.

#### Résultat Long

*Pas de résultat long pour ce check.*

#### Description des erreurs

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 au 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 ]

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 [194.168.1.107]
  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 [194.168.1.107]
  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 des architectures.

```
=> Arbiters CONFLICT
Architecture List :

1. Production [194.168.1.107]
  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 [194.168.1.107]
  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 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 cohérence des données de supervision.

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

## 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/"

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 bloquées 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.

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.

## Le démon a chargé une configuration enregistrée

Lorsque le module *last-configuration-recorder* est actif, pour les démons de type

- Broker,
- Poller,
- Reactionner,
- Receiver et
- Scheduler,

Au démarrage, le démon charge la précédente configuration qu'il a reçue de l'Arbiter, en attendant que ce dernier le contacte.

- Il permet un redémarrage du démon **même si l'Arbiter ne peut pas joindre le démon** ( *ex : coupure réseau* ),
- Cela permet aussi de rendre le **redémarrage du démon plus rapide**, dans le cas où l'Arbiter **n'a pas changé de version** de configuration.

Un **WARNING** indique que le démon n'a pas encore été contacté par l'Arbiter.

```
Arbiters connection:

Architecture List :

1. Shinken-Production [loaded from retention]
  o [WARNING] arbiter-master : loaded from retention 52s ago [waiting for Arbiter connection].
```

## Erreur de vol de CPU

Seulement si la machine virtuelle est hébergée sur un hyperviseur VMWare

- Si la VM se fait voler trop de temps de calcul (CPU Stolen), le check sera mis en **WARNING** ou en **CRITIQUE** ( *en fonction du taux de vol fixé par défaut ou indiqué par l'utilisateur* ).
  - on peut avoir plus d'informations sur cet indicateur et comment réduire la perte 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
```

## Métriques

Nom	Unité	Description
get_lock_time	seconde	Temps de connexion et d'obtention d'un appel bloquant dans le démon afin de voir si les appels bloquants ne sont pas trop longs

