

# Synchronizer - Authentification unique ( SSO )

## Sommaire

[Introduction](#)  
[Configuration](#)

## Contexte

Le modèle shinken-reactionner vous permet de superviser un hôte hébergeant le démon Reactionner ( voir la page [Reactionner](#) ).

## Description du modèle

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

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

## Checks

### Reactionner - \$KEY\$ - Running Well

Vérifie que le Reactionner est joignable sur le réseau, affiche son numéro de version, ses tags et le statut de connexion avec les Schedulers

? Unknown Attachment

### Reactionner - \$KEY\$ - Performance

Affiche les statistiques des performances de l'exécution des checks dans le Reactionner.

? Unknown Attachment

Remarque : Si le Reactionner est en exécution sur une machine virtuelle supervisé par VMware, alors le pourcentage de temps de vol de CPU ( *CPU Ready* ) sera affiché.

## Données du modèle

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

Les données suivantes sont disponibles pour le Reactionner:

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 Reactionner	http	shinken
CHECK_SHINKEN_TIMEOUT	Timeout utilisé pour établir la connexion avec le Reactionner	3	shinken
REACTIONNER_PORT	Port utilisé pour établir la connexion avec le Reactionner	7769	Locale
REACTIONNER_LIST	Liste de Reactionner ( <i>Multi-démon</i> )	reactionner-master\$( \$_HOSTREACTIONNER_PORT\$ )\$	Locale - <a href="#">Duplicate For Each</a>
NB_CHECK_IN_TIMEOUT_TOLERATE	Nombre de checks en timeout provoquant une sortie en erreur du check	0	Locale
REACTIONNER_NB_CHECK_IN_TIMEOUT_TOLERATE	Nombre de checks en timeout provoquant une sortie en erreur du check	\$_HOSTNB_CHECK_IN_TIMEOUT_TOLERATES\$	Locale
ACTIVE_REACTIONNER_LATENCY	Latence de connexion ( <i>en secondes</i> ) au-delà de laquelle le check sort en erreur	0.5	Locale

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

## 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		
Reactionner - \$KEY\$ - Performance	cpu_usage	Utilisation du CPU par le démon		
Reactionner - \$KEY\$ - Performance	nb_action_done_per_sec	Nombre de notifications & event handlers exécutés par seconde en moyenne sur la dernière minute		
Reactionner - \$KEY\$ - Performance	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)		
Reactionner - \$KEY\$ - Running Well	nb_check_in_timeout	Nombre de notifications & event handlers qui ont dépassés leur temps de timeout au cours de la dernière minute		

## Commandes

Nom du check	Commande du check	Ligne de commande
Reactionner - \$KEY\$ - Performance	check_shinken_reactionner!stats!\$VALUE1\$	\$PLUGINSDIR\$/check_shinken -H "\$HOSTADDRESS\$" -p "\$ARG2\$" --shinkenversion "\$SHINKENVERSION\$" -t reactionner -m \$ARG1\$ --active_reactionner_latency "\$_HOSTACTIVE_REACTIONNER_LATENCY\$" --check_tolerate "\$_HOSTNB_CHECK_IN_TIMEOUT_TOLERATE\$" --timeout \$_HOSTCHECK_SHINKEN_TIMEOUT\$ -w \$_HOSTTHRESHOLD_CPU_STOLEN_WARNING\$ -c \$_HOSTTHRESHOLD_CPU_STOLEN_CRITICAL\$
Reactionner - \$KEY\$ - Running Well	check_shinken_reactionner!alive!\$VALUE1\$	\$PLUGINSDIR\$/check_shinken -H "\$HOSTADDRESS\$" -p "\$ARG2\$" --shinkenversion "\$SHINKENVERSION\$" -t reactionner -m \$ARG1\$ --active_reactionner_latency "\$_HOSTACTIVE_REACTIONNER_LATENCY\$" --check_tolerate "\$_HOSTNB_CHECK_IN_TIMEOUT_TOLERATE\$" --timeout \$_HOSTCHECK_SHINKEN_TIMEOUT\$ -w \$_HOSTTHRESHOLD_CPU_STOLEN_WARNING\$ -c \$_HOSTTHRESHOLD_CPU_STOLEN_CRITICAL\$

Les modes dépréciés ("-m") :

- api\_connection
- cpu\_load
- overload\_protection

## Check : Reactionner - \$KEY\$ - Running Well

### Description des erreurs

#### Problème de surcharge des disques constaté lors de l'écriture 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

#### 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 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 cohérences des données de supervision.

## ? Unknown Attachment

### 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 Arbiter 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

**i** 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

## ? Unknown Attachment

### 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.

## ? Unknown Attachment

## Check : Reactionner - \$KEY\$ - Performance

### Description des erreurs

#### Erreur de vol de CPU

Seulement si votre machine virtuelle est hébergé 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 ).
  - 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 \(%ready + %costop\)](#)

? Unknown Attachment

? Unknown Attachment

#### 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

? Unknown Attachment

#### 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.

? Unknown Attachment