

Connection Failed SSH

Contexte

Le modèle shinken-scheduler vous permet de superviser un hôte hébergeant le démon [Scheduler](#).

Sommaire

- [Contexte](#)
- [Paramétrage](#)
 - [Données utilisées provenant du modèle](#)
 - [Données utilisées provenant du check](#)
- [Résultat](#)
 - [Exemple](#)
 - [Interprétation des données](#)
- [Métriques](#)
- [Mise en place \(pré-requis pour ce check \)](#)

Description du modèle

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

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

Checks

Nom du check	Description	Exemple de sortie
Scheduler - \$KEY\$ - Running Well	Vérifie que le démon Scheduler peut être correctement contacté sur le réseau et affiche son numéro de version. Affiche également le nombre d'éléments qu'il gère dans un tableau, en distinguant les hôtes, les Clusters et les Checks. Vous aurez également le Total des éléments. (Résultat court) Vérifie également que les modules sont opérationnels (Résultat long).	? Unknown Attachment
Scheduler - \$KEY\$ - Performance	Affiche l'utilisation CPU ainsi que les statistiques des Pollers et des Reactionners qu'il gère. Si des limites de CPU ou de RAM sont atteintes sur vos Pollers, alors ces limites seront affichées. Si certains de vos Pollers sont en spare, ils seront également affichés, avec le texte (SPARE) après leurs noms. Le retour du check Scheduler - Performance contient également un tableau de classification des checks ordonnancés par le Scheduler.	? Unknown Attachment ? Unknown Attachment

Paramétrage des checks

Les checks du Scheduler peuvent être configurés via des données fournies par le modèle.. Les paramètres SSH sont décrits sur la page [Réention Mongodb](#)

Les données suivantes sont disponibles pour le Scheduler:

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 Scheduler	http	shinken
CHECK_SHINKEN_TIMEOUT	Timeout utilisé pour établir la connexion avec le Scheduler	3	shinken
SCHEDULER_PORT	Port utilisé pour établir la connexion avec le scheduler	7768	Locale
SCHEDULER_LIST	Liste de Scheduler (Multi-démon)	scheduler-master\$(\$_HOSTSCHEDULER_PORT\$)\$	Locale - Duplicate For Each
PASSIVE_POLLER_LATENCY	Latence de connexion (en secondes) au-delà de laquelle le check sort en erreur	0.5	Locale

Données de performances

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

Nom du check	Nom de la métrique	Explication
Scheduler - \$KEY\$ - Running Well	nb_check	Nombre de checks gérés par ce Scheduler.
Scheduler - \$KEY\$ - Running Well	nb_clusters	Nombre de clusters gérés par ce Scheduler.
Scheduler - \$KEY\$ - Running Well	nb_hosts	Nombre d'hôtes gérés par ce Scheduler.
Scheduler - \$KEY\$ - Running Well	nb_late	Nombre d'éléments en retard de lancement (late) dans ce Scheduler
Scheduler - \$KEY\$ - Performance	average_scheduler_cpu_estimated_overload	Estimation de la surcharge du Scheduler. <ul style="list-style-type: none"> Si cette métrique est à 0 alors le Scheduler n'est pas en surcharge. Si cette métrique est trop souvent supérieure à 0, c'est l'indication qu'il y a besoin d'un Scheduler supplémentaire.
Scheduler - \$KEY\$ - Performance	average_scheduler_cpu_usage	Durée d'un cycle de traitement du Scheduler. <ul style="list-style-type: none"> Le maximum est à 100. Plus cette valeur est haute plus cela indique une charge sur le Scheduler.
Scheduler - \$KEY\$ - Performance	checks_todo_by_sec	Nombre de vérifications d'hôtes et de checks générées par seconde dans le Scheduler (moyenne glissante calculée sur 1min).
Scheduler - \$KEY\$ - Performance	checks_done_by_sec	Nombre de résultats de vérification d'hôtes et de checks donnés par les Pollers par seconde (moyenne glissante calculée sur 1min).
Scheduler - \$KEY\$ - Performance	notifications_todo_by_sec	Nombre de notifications générées par seconde dans le Scheduler (moyenne glissante calculée sur 1min).
Scheduler - \$KEY\$ - Performance	notifications_done_by_sec	Nombre de notifications fait par les Reactionners par seconde (moyenne glissante calculée sur 1min).
Scheduler - \$KEY\$ - Performance	nb_poller	Nombre de Pollers connectés à ce Scheduler.
Scheduler - \$KEY\$ - Performance	nb_poller_in_overload	Nombre de Pollers connectés à ce Scheduler en surcharge.
Scheduler - \$KEY\$ - Performance	nb_reactionner	Nombre de Reactionner connectés à ce Scheduler.
Scheduler - \$KEY\$ - Performance	nb_reactionner_in_overload	Nombre de Reactionner connectés à ce Scheduler en surcharge.

Détail des commandes

Nom du check	Commande du check	Ligne de commande
Scheduler - \$KEY\$ - Performance	check_shinken_scheduler!stats!\$VALUE1\$	\$PLUGINDIR\$/check_shinken -H "\$HOSTADDRESS\$" -p "\$ARG2\$" --shinkenversion "\$SHINKENVERSION\$" -t scheduler -m \$ARG1\$ -l "lck-\$LASTSERVICECHECK\$" --passive_poller_latency "\$_HOSTPASSIVE_POLLER_LATENCY\$" --timeout \$_HOSTCHECK_SHINKEN_TIMEOUT\$

Scheduler - \$KEY\$ - Running Well	check_shinken_scheduler!alive!\$VALUE1\$	\$PLUGINDIR\$/check_shinken -H "\$HOSTADDRESS\$" -p "\$ARG2\$" --shinkenversion "\$SHINKENVERSION\$" -t scheduler -m \$ARG1\$ -I "lck-\$LASTSERVICECHECK\$" --passive_poller_latency "\$_HOSTPASSIVE_POLLER_LATENCY\$" --timeout \$_HOSTCHECK_SHINKEN_TIMEOUT\$
------------------------------------	--	---

Les modes dépréciés ("-m") :

- api_connection
- late_checks
- latency
- top10_average
- top10_total

Interprétation des données de l'état de santé du Scheduler

Statistiques de l'état de santé du Scheduler

Si votre Scheduler est en bon état de fonctionnement, c'est-à-dire qu'il permet d'ordonnancer correctement tous les checks et de recevoir les résultats de ces checks en temps et en heure, alors le statut du retour du check "Scheduler - Running Well" est OK.

Suite à ce retour, en plus du numéro de version du Scheduler, un tableau est affiché, il contient l'ensemble des éléments que gère le Scheduler.

Enfin, le royaume du Scheduler est affiché.

? Unknown Attachment

Description des erreurs de Scheduler - \$KEY\$ - Running Well

Erreur de surcharge des disques de logs

Erreur	Description de l'erreur	Affichage
Disque des logs trop lent	En cas de disques trop lents sur le volume des logs, le check sera mis en WARNING avec l'erreur suivante.	? Unknown Attachment

Exemple d'un état de santé dégradé du Scheduler

Il peut arriver que vos Pollers ne permettent pas d'absorber tous les checks ordonnancés par le Scheduler, et dans ce cas, certains checks seront en retard!

L'état de retour du check est alors Warning et le nombre de checks en retard est affiché, avec son pourcentage basé sur le nombre total de checks gérés par le Scheduler.

Le nombre de checks en retard est ensuite affiché, groupé par [Poller Tag](#).

Le temps d'attente moyen d'un check sur le Scheduler avant d'être récupéré par un Poller pour traitement est également affiché. Dans le cas de checks en retard, il peut être intéressant de comparer cette statistique avec d'autres Scheduler de votre architecture (si vous pouvez).

Enfin, le check en **WARNING** rappelle à quel royaume appartient le Scheduler.

? Unknown Attachment

Interprétation des statistiques de performance du Scheduler

La supervision d'un démon Scheduler présente un grand nombre de statistiques de performances qui permettent de visualiser le travail d'ordonnancement effectué par le Scheduler ainsi que les statistiques des Poller qui viennent se connecter sur celui-ci.

Le démon Scheduler va effectuer tout le travail d'ordonnement, et c'est à lui que vont s'adresser (en autres) les démons Poller et Reactionner pour récupérer les checks et les notifications à effectuer. Son bon fonctionnement est donc vital au bon fonctionnement de votre architecture Shinken. Aussi, pour dimensionner correctement une installation Shinken Entreprise, il est important de pouvoir visualiser combien de checks ses Poller satellites peuvent traiter, ainsi que leurs utilisations CPU et RAM.

Les checks du Scheduler fournis dans le pack Shinken fournissent donc un grand nombre de données sur les performances du Scheduler et de ses Pollers.

L'ensemble des informations se retrouve dans le résultat court du check.

Statistiques générales des performances du Scheduler

La première statistique remontée par le check est le pourcentage CPU moyen utilisé par le démon Scheduler sur le serveur supervisé.

La deuxième statistique remontée est le temps d'attente moyen d'un check sur le Scheduler avant d'être récupéré par un Poller pour traitement.

Information générale des Satellites du Scheduler

Suite aux statistiques générales, un premier tableau rassemble les données de performance des satellites du Scheduler de type Poller.

La première partie du tableau (les trois premières colonnes) identifie les Pollers, en affichant leurs noms, leurs appartenances à un Royaume, et enfin leurs tags (None si aucun tag n'est associé au Poller).

Statistiques des checks

Les deux colonnes suivantes affichent les performances de traitement des checks des Pollers :

- checks todo : Moyenne du nombre de checks à **traiter** par le Poller (en checks par seconde)
- checks done : Moyenne du nombre de checks **traités** par le Poller (en checks par seconde)

Vous pourrez donc avoir l'information du nombre de checks récupérés et traités par vos différents Pollers rattachés à ce Scheduler et ainsi pouvoir comparer les performances de vos Pollers suivant leur positionnement dans votre architecture réseau, ou suivant leur puissance matérielle.

Utilisation du CPU

CPU Available

La colonne "CPU available" concerne les performances CPU des Pollers.

Cette information représente la charge du Poller. Il s'agit d'un indicateur général indiquant si le Poller peut encore supporter des checks supplémentaires, ou s'il est chargé au maximum. Cet indicateur n'est pas lié aux autres indicateurs de performances de la machine (File d'attente CPU, mémoire)

Une pastille orange précédant la mention "Poller load" signifie que le Poller ne peut plus prendre de checks supplémentaires.

C'est donc un signe indiquant qu'il faudrait ajouter un Poller supplémentaire dans l'architecture Shinken.

? Unknown Attachment

? Unknown Attachment

? Unknown Attachment

? Unknown Attachment

Si tous vos Pollers sont en surcharge, alors les checks ne pourront plus être récupérés, et vous aurez des retards visibles dans le retour de votre check "Scheduler - Running Well".

Il vous faudra de toute urgence rajouter des Pollers dans votre royaume.

Voici par exemple une surcharge d'un Poller.

Metrics: CPU used by the poller

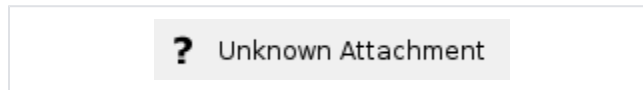
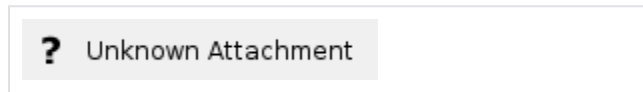
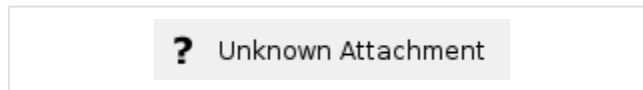
La colonne "Metrics: CPU used by the poller" permet d'afficher la consommation CPU utilisé par le Poller. Comme son nom l'indique, un graphique est associé à ce check et permet d'afficher cette métrique.

Lorsque le Poller utilise le maximum de CPU possible sur le serveur, une information apparaît.

Cette valeur de CPU utilisée par le Poller ne sera jamais à 100%, car le système Linux hébergeant le démon utilise une partie du CPU, comme les applications additionnelles que ce serveur peut utiliser.

Plus il y a d'application sur votre serveur Poller consommant du CPU, moins votre démon pourra utiliser de CPU à ses fins, et atteindra rapidement sa charge maximale utilisable (bien en deçà de 100%).

Lorsque la limite est atteinte, voici l'affichage dans le tableau.



Utilisation de la RAM

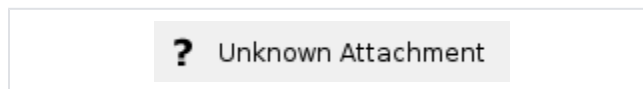
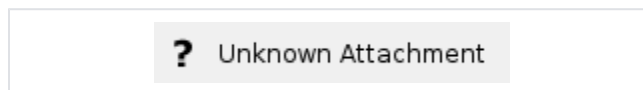
% used RAM on the server

La dernière colonne du tableau représente le pourcentage de RAM utilisé sur le serveur.

Si la valeur détectée est inférieure à la limite définie, alors la consommation est considérée comme normale et la pastille "normal" est affichée.

La limite paramétrée dans le Poller est affichée entre parenthèses.

Si l'utilisation de la mémoire (RAM) sur le serveur dépasse le seuil défini dans la configuration de ce Poller, une pastille rouge de dépassement est affichée, indiquant l'utilisation excessive de la mémoire. Lorsque cet avertissement est affiché, le Poller n'exécute plus de checks supplémentaires tant que l'utilisation de la mémoire est supérieure au seuil défini.

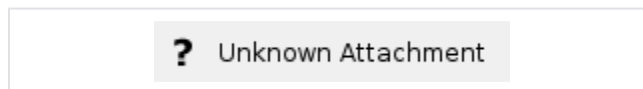


Load

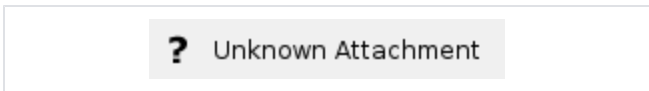
Si par exemple le CPU n'est pas utilisé au maximum de ses performances, mais que sa running queue (file d'attente) est importante, la limitation de CPU ne peut prévenir ce cas.

Pour être sur que le Poller ne tente d'exécuter des checks sur une machine surchargée le Poller se limitera en fonction de l'état de la file d'attente processeur (représentant la valeur source du load average).

Dans ce cas, le Poller n'exécutera plus de checks supplémentaires tant que le nombre de processus dans la file d'attente du processeur sera supérieur au seuil choisi. La limite paramétrée dans le Poller est affichée entre parenthèses.

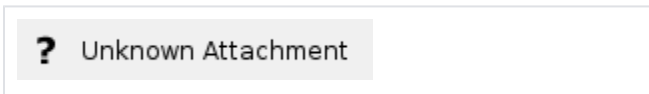


Lorsque la limite est atteinte pour ce Poller, alors le check "Scheduler - Performance" ajoute une pastille rouge vous informant du dépassement de la limite.



Information des Reactionners Satellites

Suite aux statistiques des Satellites de type "Poller", un deuxième tableau rassemble les données de performance des satellites du Scheduler de type Reactionner.



Les trois premières colonnes représentent, comme pour le tableau précédent, les données d'identification des Reactionners venant récupérer les notifications auprès du Scheduler.

Les deux colonnes suivantes permettent d'obtenir les statistiques des notifications à traiter par les Reactionner ainsi que les notifications déjà réalisées (en nombre de notifications par seconde).

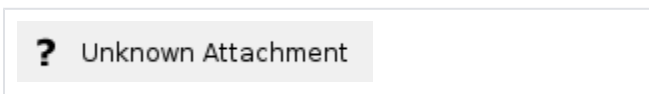
Enfin les deux dernières colonnes affichent les informations CPU des Reactionners, de la même manière que pour les Pollers.

Classification des checks

Le Scheduler est un ordonnanceur de checks.

Cet ordonnancement peut être fait pour différentes raisons qui sont énumérées dans la colonne "Causes" de ce tableau ci-contre :

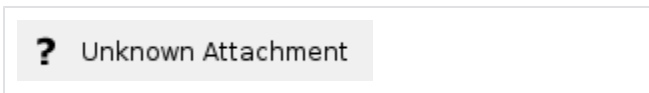
- **Dependency** : Les checks qui sont demandés, car liés à une dépendance (checks liés à son hôte ou hôte fils lié à son hôte parent)
- **Retry** : Les checks qui sont revérifiés pour la confirmation des états, via la propriété "Intervalle de nouvelles tentatives de confirmations d'état" des checks et des hôtes
- **Force** : Les checks qui sont demandés par les utilisateurs depuis l'interface de visualisation (bouton "Forcer la vérification")
- **Schedule** : Les checks qui sont ordonnancés de manière régulière via la propriété "intervalle entre les vérifications" des checks et des hôtes (normalement le plus actif des 4 raisons)



Pour chaque raison, le nombre de checks par seconde est affiché dans la deuxième colonne.

Consommation de temps CPU des checks

Le check "Scheduler - Performance" peut également détecter si la commande d'un check prend trop de temps CPU lors de son exécution.



Si le seuil est atteint (et dans ce cas seulement), le check passe en état **"WARNING"** et le tableau ci-contre apparaît dans le résultat du check.

Ce tableau contient les noms des commandes, leurs temps CPU consommés, le seuil fixé pour cette commande et enfin la date de l'exécution.

Par défaut, le seuil est fixé à 5 secondes. Cette propriété nommée "Seuil d'alerte de l'utilisation CPU (sec)" est modifiable via l'UI de configuration dans les onglets "Supervision" des objets "hôte" et "check" et dans l'onglet "Général" des commandes. La clé d'import est **warning_threshold_cpu_usage**

Ce paramètre est aussi modifiable globalement dans le fichier **/etc/shinken/shinken.cfg**.

```
# How many seconds a command check (for hosts,
clusters and checks) is allowed to consume cpu
# before raise a warning in check scheduler
performance
# by default: 5
#warning_threshold_cpu_usage=5
```

Un redémarrage de l'Arbiter sera ici requis.

Cas Particuliers d'erreur

Si un Poller est détecté comme injoignable (par exemple s'il y a un problème réseau avec ce démon ou alors qu'il vient juste d'être désactivé depuis l'Arbiter) alors un message est affiché.

? Unknown Attachment