

Mapping des informations collectées des champs du serveur VMWare vers les propriétés et les données Shinken

Sommaire

- Concept
- Définition des règles de mapping
 - Forcer la valeur des propriétés
 - Liste des clés d'import pouvant être forcées
 - Exemple de mapping dans le fichier "user_mapping_rules_vm.json"
- Opérateur pour les règles de mapping
 - Opérateur VALUES pour les listes
 - Définition de l'utilisation de VALUES
 - Opérateur TRANSFORM pour les transformations
 - Opérateur CONCAT pour les concaténations
 - Opérateur CONCAT_ON_ALL pour les concaténations sur les listes
 - Opérateur MANY_FIELDS_BY_REGEX pour règles de mapping dynamiques (expressions régulières)
 - Exemple d'expression régulière
- Désactivation d'un mapping
- Visualiser la liste des mappings
- Exemple

Contexte

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

Description du modèle

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

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	
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'élément qu'il gère dans un tableau, en distinguant les hôtes, les Clusters et les Checks. Vous aurez également le Total des éléments. (<i>Résultat court</i>) Vérifie également que les modules sont opérationnels (<i>Résultat long</i>).	? Unknown Attachment
eScheduler - \$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. Si jamais le démon Arbitrer est en exécution sur une machine virtuelle supervisé par VMware, alors le pourcentage de temps de vol de CPU (<i>CPU Ready</i>) sera affiché.	? Unknown Attachment ? Unknown Attachment

Données du modèle

Les checks du Scheduler peuvent être configurés via des données fournies par le modèle. Les paramètres SSH sont décrits sur la page [Rétention 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 (<i>Multi-démon</i>)	scheduler-master\$(\$_HOSTSCHEDULER_PORT\$)\$	Locale - Duplicate For Each
PASSIVE_POLLER_LATENCY	Latence de connexion (<i>en secondes</i>) au-delà de laquelle le check sort en erreur	0.5	Locale
SCHEDULER_RETENTION_RETENTION-IS-TOO-OLD-AFTER_X_MINUTES	Temps (<i>en minutes</i>) additionnel de marge rajouté à l' intervalle de sauvegarde de la rétention avant qu'une rétention ancienne ne soit déclarée trop vieille et retourne en WARNING.	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 **shinken-scheduler** 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
Scheduler - \$KEY\$ - Running Well	cpu_stolen_vmware_percent_ready	(<i>Seulement si le démon est situé sur une VM VMWare</i>) 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>)
Scheduler - \$KEY\$ - Running Well	nb_checks	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 de hôte gérés par ce Scheduler.
Scheduler - \$KEY\$ - Running Well	nb_late_checks	Nombre d'exécution de checks (<i>pour les Pollers</i>) en retard de lancement (<i>late</i>) dans ce Scheduler.
Scheduler - \$KEY\$ - Running Well	nb_late_notifications	Nombre d'exécution de notifications (<i>pour les Reactionners</i>) en retard de lancement (<i>late</i>) dans ce Scheduler.
Scheduler - \$KEY\$ - Running Well	nb_late_event_handlers	Nombre de gestionnaires d'événement à exécuter (<i>pour les Reactionners</i>) en retard de lancement (<i>late</i>) 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 (<i>moyenne glissante calculée sur 1mn</i>).
Scheduler - \$KEY\$ - Performance	checks_done_by_sec	Nombre de résultats de vérification d'hôtes et de checks donnés par les Pollers par seconde (<i>moyenne glissante calculée sur 1mn</i>).
Scheduler - \$KEY\$ - Performance	notifications_todo_by_sec	Nombre de notifications générées par seconde dans le Scheduler (<i>moyenne glissante calculée sur 1mn</i>).
Scheduler - \$KEY\$ - Performance	event_handlers_todo_by_sec	Nombre de gestionnaires d'événement lancés par seconde dans le Scheduler (<i>moyenne glissante calculée sur 1mn</i>).
Scheduler - \$KEY\$ - Performance	load_retention_time	Durée en seconde du dernier chargement de rétention

Scheduler - \$KEY\$ - Performance	notifications_and_event_handlers_done_by_sec	Nombre de notifications & de gestionnaires d'événement fait par les Reactionners par seconde (<i>moyenne glissante calculée sur 1mn</i>).
Scheduler - \$KEY\$ - Performance	nb_pollers	Nombre de Pollers connectés à ce Scheduler.
Scheduler - \$KEY\$ - Performance	nb_pollers_in_overload	Nombre de Pollers connectés à ce Scheduler en surcharge.
Scheduler - \$KEY\$ - Performance	nb_reactionners	Nombre de Reactionners connectés à ce Scheduler.
Scheduler - \$KEY\$ - Performance	nb_reactionners_in_overload	Nombre de Reactionners connectés à ce Scheduler en surcharge.
Scheduler - \$KEY\$ - Performance	save_retention_time	Durée en seconde de la dernière sauvegarde de rétention.

Commandes

Nom du check	Commande du check	Ligne de commande
Scheduler - \$KEY\$ - Performance	check_shinken_scheduler!status!\$VALUE1\$	\$PLUGINSDIR\$/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\$ -w \$_HOSTTHRESHOLD_CPU_STOLEN_WARNING\$ -c \$_HOSTTHRESHOLD_CPU_STOLEN_CRITICAL\$ --scheduler_too_old_retention_save_margin \$_HOSTSCHEDULER__RETENTION__RETENTION-IS-TOO-OLD-AFTER_X_MINUTES\$
Scheduler - \$KEY\$ - Running Well	check_shinken_scheduler!alive!\$VALUE1\$	\$PLUGINSDIR\$/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\$ -w \$_HOSTTHRESHOLD_CPU_STOLEN_WARNING\$ -c \$_HOSTTHRESHOLD_CPU_STOLEN_CRITICAL\$ --scheduler_too_old_retention_save_margin \$_HOSTSCHEDULER__RETENTION__RETENTION-IS-TOO-OLD-AFTER_X_MINUTES\$

Les modes dépréciés sur les commandes ("-m") :

- api_connection
- late_checks
- latency
- top10_average
- top10_total

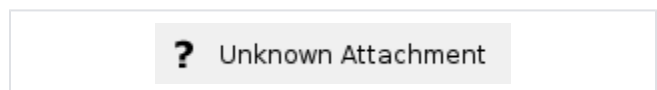
Interprétation du check Scheduler - \$KEY\$ - Running Well

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.



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

Problème de surcharge des disques de logs

Erreur	Description de l'erreur	Affichage
--------	-------------------------	-----------

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

Etat 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 seront en retard! Les checks sont considérés en retard s'ils dépassent 10 secondes à partir du moment où le Scheduler les a mis à disposition des Pollers.

L'état de retour du check "Scheduler - \$KEY\$ - Running Well" est alors **WARNING** et le nombre de checks en retard avec le pourcentage des checks en retard par rapport au volume géré par le Scheduler sont affichés.

Le volume géré par le Scheduler est calculé à partir:

- Du nombre de checks
- Du nombre d'hôtes (*un check est fait pour chaque commande de vérification*)
- Du nombre de clusters (*un check est fait pour chaque définition du cluster*)

Le nombre de checks en retard est ensuite affiché, groupé par Poller Tag.

Pour les checks qui ont été exécutés durant les 5 dernières minutes, le temps d'attente moyen sur le Scheduler avant d'être récupéré par un Poller est également affiché. Si ce chiffre s'approche des 10 secondes :

- Vos Pollers n'arrivent pas à absorber toute la charge : il peut être nécessaire d'ajouter un nouveau Poller.
- Il peut également s'agir d'un problème de latence réseau ou que l'un des Pollers ne soit plus disponible.

Ces informations sont également disponibles pour les notifications et les événements. Dans ce cas, le démon concerné est le Reactionner

Enfin, le check rappelle à quel royaume appartient le Scheduler.

Interprétation du check Scheduler - \$KEY\$ - Performance

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 tous le travail d'ordonnancement, 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

- 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.
- **JROY: Manque descriptif rétention**
- Si votre machine virtuelle est hébergée sur un système VVMare alors une dernière statistique sera remontée.
 - Elle affiche le taux de CPU %ready (*vol de temps de calcul du CPU votre machine par les autres machines virtuelles de l'hyperviseur*).

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 fixé par défaut ou indiqué par l'utilisateur*) avec l'erreur suivante.

? Unknown Attachment

? Unknown Attachment

? Unknown Attachment

JROY: Screen pas bon

? Unknown Attachment

? Unknown Attachment

Suivi des chargements/sauvegardes des données de rétention

Les données de rétention sont chargées/sauvegardées par les démons Schedulers. Un affichage permet de voir:

- de quand date et a duré le dernier chargement de rétention (lors d'une nouvelle configuration)
- de quand date et a duré la dernière sauvegarde de rétention (lors d'une nouvelle configuration, ou alors toutes les `retention_interval` disponible dans le fichier `shinken.cfg`)

Si la dernière sauvegarde de rétention est trop vieille (plus que `retention_interval + SCHEDULER__RETENTION__RENTENTION-IS-TOO-OLD-AFTER_X_MINUTES`), alors un WARNING sera remonté

Si le dernier essai en date de sauvegarde de rétention est en ERROR, alors un message sera disponible avec le dernier message du module en question

? Unknown Attachment

? Unknown Attachment

? Unknown Attachment

? Unknown Attachment

Suivie des Pollers Satellites

Informations générales

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 à faire / fait

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

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

? Unknown Attachment

? Unknown Attachment

Utilisation du CPU

CPU Available

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

? Unknown Attachment

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 si 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.

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.

? Unknown Attachment

CPU used by the poller

La colonne "CPU used by the poller" permet d'afficher la consommation CPU utilisée 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.

? Unknown Attachment

? Unknown Attachment

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 parenthèse.

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.

? Unknown Attachment

? Unknown Attachment

Load

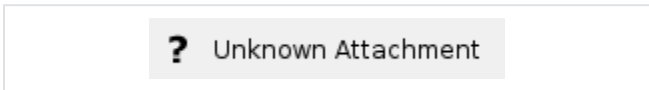
Si par exemple le CPU n'est pas utilisé au maximum de ses performances mais que sa file d'attente (*running queue*) est importante, la limitation de CPU ne peut prévenir ce cas.

? Unknown Attachment

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

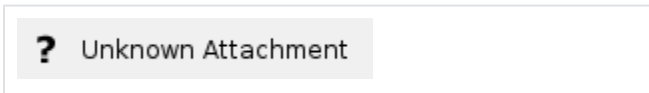
Dans ce cas, le Poller n'exécutera plus de checks supplémentaires tant que le nombre de processus dans la file d'attente du processeur sera supérieur au seuil choisi. La limite paramétrée dans le Poller est affichée entre parenthèse.

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



Suivie 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 précédant, les données d'identification des Reactionners venant récupérer les notifications auprès du Scheduler.

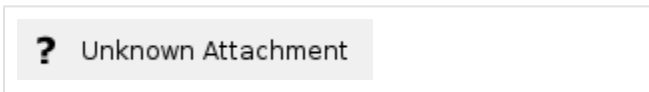
Les deux colonnes suivantes permet d'obtenir les statistiques des notifications à traiter par les Reactionner ainsi que les notifications déjà réalisées (*en nombre de notifications par seconde*).

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

Type de checks fait par seconde

Le Scheduler est un ordonnanceur de checks.

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

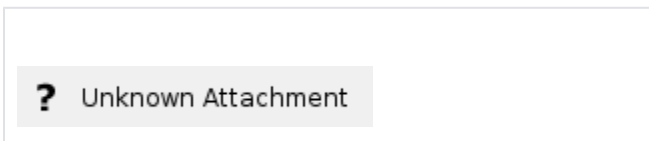


- **Dependency** : Les checks qui sont demandés car liés à une dépendance (*checks liés à son hôte ou hôte fils liés à son hôte parent*)
- **Retry** : Les checks qui sont revérifiés pour la confirmation des états, via la propriété "Intervalle de nouvelles tentatives de confirmations d'état" des checks 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 actifs 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 prends 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