

La surcharge des propriétés pour un check

Sommaire

- Principe
- Accéder à la surcharge
 - Activer le mode surcharge
 - Ouvrir le formulaire de surcharge
- Présentation visuelle
- Surcharger une propriété
 - Réaliser une surcharge
 - Revenir à la valeur initiale du check
 - Cas spécifiques
- Les compteurs
 - Les compteurs de surcharge
 - Les compteurs d'avertissement et d'erreur
- La surcharge dans les modèles d'hôte
 - Affichage dans la surcharge sur l'hôte
- Surcharge et héritage additif
- La surcharge et l'essai des checks
- Surcharge inactive
 - Rattacher une surcharge inactive
 - Compteur de surcharge inactive
 - Supprimer une surcharge inactive
- Surcharges au format Cfg

Le healthcheck peut afficher de nombreux cas d'erreur différents. Aussi, certaines informations importantes sont présentées régulièrement.

Pour faciliter l'utilisation du Healthcheck, cette section présente ces différentes informations et erreurs courantes, accompagnées de leur interprétation.

- [Principe](#)
- [Accéder à la surcharge](#)
- [Présentation visuelle](#)
- [Surcharger une propriété](#)
- [Les compteurs](#)
- [La surcharge dans les modèles d'hôte](#)
- [Surcharge et héritage additif](#)
- [La surcharge et l'essai des checks](#)
- [Surcharge inactive](#)
- [Surcharges au format Cfg](#)

Le démon est configuré avec l'adresse "localhost"

Puisque les démons constituant une installation de Shinken Entreprise ont pour vocation à être utilisés dans une architecture distribuée, il ne faut pas définir les démons avec des adresses locales. Les démons vont devoir communiquer entre eux, il faut donc les configurer avec une adresse précise pour éviter tout problème dans une architecture distribuée.

Cette erreur est très fréquemment rencontrée dans le healthcheck puisque dans l'installation par défaut de Shinken Entreprise, les démons utilisent l'adresse localhost. Dès que des démons seront utilisés dans la même configuration sur plusieurs machines différentes, il est impératif d'utiliser des adresses précises.

Le démon est injoignable

Le Scheduler "*scheduler-vm3*" est injoignable.

Dans ce cas, on exécute le healthcheck sur la machine vm3. On peut donc déduire que le démon est éteint.

Sur un démon hébergé sur une machine distante, il faudra d'abord déterminer si la machine est joignable avant de pouvoir affirmer que le démon est éteint. Un healthcheck local sur la machine hébergeant le démon peut confirmer cette hypothèse.

Le démon et son Arbiter ont des versions différentes

? Unknown Attachment

? Unknown Attachment

Pour garantir un bon fonctionnement, il faut que tous les démons présents dans la configuration possèdent la même version de Shinken Entreprise.

Dans ce cas, le poller "*poller-domtom*" et l'Arbiter n'ont pas la même version installée, ce qui peut provoquer des dysfonctionnements.

Conflit d'Arbiter sur un démon

Lors d'une mauvaise configuration, il arrive que 2 installations différentes de Shinken Entreprise utilisent les démons du même serveur.

Dans ce cas, plusieurs Arbiters envoient une configuration contradictoire au même démon. Dans ce cas, une erreur est affichée dans le healthcheck indiquant le conflit d'Arbiter.

Pour chaque Arbiter, le nom et l'adresse de l'Arbiter concerné sont également indiqués.

La configuration de l'Arbiter n'a pas été trouvé

Ce message apparaît quand la commande healthcheck utilise l'option --locale ou -l et que la propriété *host_name* n'a été renseignée dans aucun fichier de configuration de ou des Arbiter(s).

Pour résoudre ce problème, il faut dans le fichier de configuration de l'Arbiter qui est présent sur cette machine (situé dans */etc/shinken/arbiter(s)*), changer la propriété *host_name* avec le nom de la machine (qui est trouvable grâce à la commande *hostname*)

Décalage de temps entre 2 démons

Puisque Shinken Entreprise repose sur une architecture distribuée composée de plusieurs démons, la synchronisation de l'horloge entre les démons est un point important pour un bon fonctionnement du système.

Dans le Healthcheck, si un démon ne possède pas la même heure que l'Arbiter, une erreur est affichée et indique le décalage horaire.

Dans l'exemple, le poller "*poller-domtom*" possède un décalage horaire de 4160 secondes avec l'Arbiter "*arbiter-vm3*".

Un démon est configuré en tant que Spare

Un démon peut être configuré en tant que Spare. Un démon Spare est par défaut inactif, et devient actif lorsqu'un démon principal n'est plus disponible, afin d'assurer une continuité du service.

Pour pouvoir identifier facilement les démons Spare dans l'architecture, une mention "SPARE" est présente à côté du nom du démon.

Si le démon prend la main afin d'assurer la continuité de service, alors une mention additionnelle "RUNNING" est affichée.

Dans le cas d'un démon Broker, une ligne supplémentaire va être affichée afin d'indiquer de quel démon master il est le spare.

Un démon master a un démon spare de désigné (seulement pour les Brokers)

Quand un démon master a un démon spare de désigné (via sa propriété *spare_daemon*) alors il sera listé.

Un démon master n'a pas de démon spare de désigné (seulement pour les Brokers)

Quand un démon master n'a pas de démon spare, ceci sera affiché.

Un démon spare qui n'a pas de démon master (seulement pour les Brokers)

Quand un démon spare n'est le spare d'aucun autre démon, il est noté comme inutilisé.

? Unknown Attachment

? Unknown Attachment

? Unknown Attachment

? Unknown Attachment

? Unknown Attachment

? Unknown Attachment

? Unknown Attachment

? Unknown Attachment

? Unknown Attachment

Informations et détections dans le cadre de virtualisation Vmware

Il se peut que vous décidiez de virtualiser votre architecture Shinken, ou simplement quelques-uns des démons, avec **VMWare**.

Si c'est le cas, une ligne sera ajoutée dans la sortie du "shinken-healthcheck" et précisera que votre satellite fonctionne sur une architecture VMWare.

Détection de la présence des "VMWare tools"

Une vérification sera faite sur la présence ou non des "VMWare tools" sur votre VM (Machine virtuelle). Si ces derniers ne sont pas installés, alors une information sera affichée.

Nous vous conseillons de toujours avoir vos "VM tools" installés, et à jour, sur l'ensemble de votre parc de VM.

Ces utilitaires peuvent être installés avec le paquet "open-vm-tools":

```
yum install open-vm-tools
```

Détection d'un pourcentage élevé de "stolen CPU"

Enfin, si votre configuration de virtualisation ne permet pas à votre machine virtuelle une utilisation correcte des CPU physiques de son hôte via les VCPU que vous avez associés à celle-ci, alors une information sera affichée. Vos VCPU n'arrivent pas à traiter assez rapidement toutes les demandes d'exécution.

La valeur "stolen CPU" en pourcentage est utilisée pour la détection. Si cette valeur est située entre 5% et 10%, alors un message de type "AT RISK" apparaît, si elle est supérieure ou égale à 10% alors un message de type "ERROR" apparaît.

Si vous recevez ce message, plusieurs options sont possibles :

- Réduisez le nombre d'allocations de VCPU sur votre VM
- Augmentez vos ressources CPU pour vos VM
- Déplacer vos VM sur un autre serveur physique moins sollicité en terme CPU

La récupération des données de VMWare sont faites via le Gatherer qui écrit le fichier

/dev/shm/vmware_stats_export.dat que chaque daemon va lire pour répondre au Shinken-healthcheck.

Si le Gatherer est éteint (il est automatiquement démarré lorsqu'on démarre un daemon shinken), alors ce fichier ne sera plus mis à jour et les informations ne seront plus fiables.

Si le fichier n'est plus mis à jour depuis plus d'une minute, alors un "AT RISK" sera affiché sur le Shinken healthcheck.

Vérifier dans les logs du Gatherer ce qui est la source du problème, et ensuite, si besoin, relancer le Gatherer sur le serveur distant via la commande:

```
service shinken-gatherer restart
```

Un module a redémarré de manière imprévue

Certains démons de Shinken Entreprise utilisent des modules qui permettent d'étendre leurs fonctionnalités. Il arrive que ces modules redémarrent pour des raisons diverses et variées.

? Unknown Attachment

? Unknown Attachment

? Unknown Attachment

? Unknown Attachment

Lorsqu'un module a redémarré dans les 2 dernières heures, un avertissement est affiché pour le module en question, car il peut s'agir d'un problème récurrent et/ou potentiellement gênant. Une analyse des logs peut permettre d'avoir plus d'informations sur les erreurs du module.

? Unknown Attachment

Erreur de communication entre les démons

Les différents démons de Shinken Entreprise communiquent entre eux pour fonctionner. Il arrive que des erreurs de communications soient remarquées et génèrent des erreurs, qui sont alors notifiées dans le Healthcheck dans la section du démon concerné.

Il est à noter que ces erreurs ne sont pas forcément fatales au bon fonctionnement de Shinken Entreprise, mais qu'elles nécessitent une attention particulière.

? Unknown Attachment

i Les erreurs de communication peuvent ne pas être graves et ne pas avoir d'incidence sur votre architecture Shinken. Cependant, si vous recevez des erreurs et que vous avez des doutes sur l'origine de ces problèmes de communication, par prévention, envoyez-nous votre log pour analyse.

Erreur d'encodage dans les fichiers de configuration d'un démon

Si les fichiers de configuration des démons situés dans `/etc/shinken` ne sont pas encodés en utf-8.

Dans l'exemple ci-contre le fichier `arbiter-master.cfg` ne respecte pas la bonne encodage (utf-8) et apparaît en erreur dans le résultat de la commande `shinken-healthcheck`.

Correction : Il vous suffit de ré-encoder le fichier au format utf-8

? Unknown Attachment

Un scheduler n'a pas de broker ou de poller ou de réagir

Les schedulers produisent des jobs à effectuer pour les pollers /réagir, ainsi que des données pour les brokers (par exemple pour l'interface de visualisation). Ils ont donc besoin de ces autres démons pour fonctionner.

En cas de mauvaise configuration de démons et/ou de royaumes, un scheduler peut ne pas être défini comme cible d'un autre démon (poller /réagir/broker).

i Si le scheduler est dans un sous royaume et qu'un poller /réagir/broker est présent dans un royaume supérieur, ceci signifie que ce dernier n'a pas l'option `manage_sub_realms` activée.

? Unknown Attachment

Un module a mis trop de temps à répondre aux demandes d'informations/statistiques

En cas de surcharge d'un module, ce dernier peut mettre trop de temps pour répondre aux demandes d'informations (2s, répétée une fois), ce qui va faire une erreur au niveau du module.

? Unknown Attachment

Erreurs de configuration concernant les données de métrologie

Dans une configuration avec plusieurs royaumes, il est plus difficile de vérifier que les données de métrologie sont bien écrites et peuvent être lues pour chaque royaume.

Il existe 2 types d'erreurs:

- Les erreurs d'écriture
- Les erreurs de lecture

Les erreurs d'écriture

Cas 1 : Les données d'un royaume ne sont pas sauvegardé

Vous êtes dans ce cas, si le message suivant apparaît :

```
[*] There is no broker configured to write data for this realm. This realm might produce data that are not stored anywhere (no Graphite backend configured)
```

Exemple : Un broker dans le royaume Corse qui n'a pas de module graphite, mais il a un module webui, il n'y a donc pas de broker qui sauvegarde les données du royaume Corse.

Correction : Ajouter le module Graphite-Perfdata sur le broker

There is no broker configured to write data for this realm. This realm might produce data that are not stored anywhere (no Graphite backend configured)

? Unknown Attachment

Cas 2 : Plusieurs brokers du même royaume écrivent dans une même base graphite

Vous êtes dans ce cas, si le message suivant apparaît :

```
[*] This realm is stored on same backends by multiple brokers : by broker * on graphite *(*) and by broker * on graphite * (*)
```

Exemple : Deux brokers sur le même royaume (Metropole) ont un même module graphite qui va écrire les données sur le même serveur graphite (172.16.0.181)

This realm is stored on same backends by multiple brokers : by broker [broker-metropole-02] on graphite backend lab-validation16 (172.16.0.181) and by broker [broker-metropole-03] on graphite backend lab-validation16 (172.16.0.181)

? Unknown Attachment

Toutes les métriques sont reçues et gérées par le broker. Chaque broker écrira donc les données dans le serveur graphite. Il y aura donc des données en double sur le serveur graphite.

Correction : Un seul broker doit écrire les données d'un royaume sur un serveur graphite. Vous pouvez :

- Enlever le module graphite_perfdata d'un des deux brokers
- créer un deuxième module graphite_perfdata qui écrira les données sur un serveur graphite différent. Afin de mettre en place de la haute disponibilité pour la base de métrologie; consultez la page [Haute disponibilité de la base de métrologie \(Graphite\)](#)

Cas 3 : Le module graphite sauvegarde uniquement des royaumes différents de celui du broker

```
Broker * has module* with realm [*] in parameter realm_store_only, but this broker handle only realms : *
```

Le broker a un module graphite_perfdata qui sauvegarde les données uniquement pour un royaume qui est différent de celui ou ceux que le broker gère. Les royaumes gérés par le module sont cités dans le paramètre "realm_store_only". Si ce paramètre est commenté ou absent, le module prend en compte tous les royaumes que le broker gère.

Correction : Il faut changer le module de type graphite_perfdata pour un autre module qui gère le/les mêmes royaumes que le broker. Il est possible de mettre un module graphite_perfdata qui prend tous les royaumes (paramètre realm_store_only commenté).

Broker broker-metropole-02 has module Graphite_Corse with realm [Corse] in parameter realm_store_only, but this broker handle only realms : Metropole

? Unknown Attachment

Cas 4 : Le paramètre realm_store_only contient un royaume qui n'existe pas

```
Module Graphite_Corse has realm [*] in parameter realm_store_only but this realm is unknown
```

Le paramètre "realm_store_only" contient un royaume qui n'est pas défini dans la configuration Shinken.

Correction : vérifiez qu'il n'y a pas d'erreurs dans le ou les noms de royaumes qui sont définis dans le paramètre "realm_store_only" dans le module graphite_perfdata cité dans la ligne d'erreur.

Les erreurs de lecture

Cas 1 : La configuration du module webui.cfg précise un royaume qui n'existe pas

Vous êtes dans ce cas, si le message suivant apparaît :

```
Realm [*] : This realm is an unknown realm. Know realms are : *
```

Dans notre exemple : Dans le fichier `/etc/shinken/module/webui.cfg` la configuration précise un royaume qui n'existe pas.

```
graphite_backends corse:192.168.1.53, Reunion:192.168.1.148, Metropole:172.16.0.181
```

Correction : Préciser un nom de royaume qui existe.

```
graphite_backends Corse:192.168.1.53, Reunion:192.168.1.148, Metropole:172.16.0.181
```

Cas 2 : La configuration du module webui.cfg précise un royaume non géré par le broker

Vous êtes dans ce cas, si le message suivant apparaît :

```
Realm [*] : This realm can't be accessed by the broker *. It can access to : *
```

Il y a deux cas possibles :

Exemple 1 : le broker ne gère pas les sous royaume, mais dans la configuration du module webui est précisée l'adresse d'un royaume fils

Exemple 2 : le broker broker-reunion reçoit un module Webui_Reunion est configuré pour le royaume corse dont il ne s'occupe pas.

Correction :

- le royaume qui ne peut être accessible est un sous royaume du royaume qui est géré par le broker. Dans ce cas, il suffit de dire au broker de gérer les sous-royaumes : ajouter la gestion des sous-royaumes par le broker dans `/etc/shinken/brokers/broker-master3.cfg`

```
manage_sub_realms 1
```

- Le royaume n'est pas un sous royaume, il faut enlever le royaume non géré du module webui. Il est possible de spécifier le caractère * dans la définition des backend afin de dire que ce module webui prendra tous les royaumes et sous-royaumes du broker vers le même graphite_backend

Cas 3 : Le serveur graphite précisé dans la configuration de webui ne gère pas les données de ce royaume

Vous êtes dans ce cas, si le message suivant apparaît :

```
Module Graphite_Corse has realm [corse] in parameter realm_store_only but this realm is unknown
```

? Unknown Attachment

```
Realm * is an unknown realm
```

? Unknown Attachment

```
Realm * can't be accessed by the broker *
```

? Unknown Attachment

```
Realm [*] : The Graphite server on * does not
contain data for this realm
```

Exemple : Le serveur graphite qui est sur 192.168.1.53 sauvegarde uniquement les données du royaume Corse. Cependant, le broker broker-metropole-01 est configuré avec une webui qui va lire les données pour le royaume Metropole sur le serveur graphite 192.168.1.53.

```
graphite_backends      Corse:192.168.1.53,
Reunion:192.168.1.148, Metropole:192.168.1.53
```

Correction : Préciser un serveur qui gère les données du royaume Metropole:

```
graphite_backends      Corse:192.168.1.53,
Reunion:192.168.1.148, Metropole:172.16.0.181
```

```
Realm [*] : The Graphite server on * does not contain data for
this realm
```

? Unknown Attachment

Cas 4 : Le serveur graphite précisé dans la configuration de webui n'est pas un serveur graphite connu

Vous êtes dans ce cas, si le message suivant apparaît :

```
The server on * is not a known Graphite server :
There is no module graphite_perfdata that writes
data on backend *
```

Exemple : La configuration de webui précise que les données du royaume Metropole sont sur 172.16.0.182 qui n'est pas un serveur graphite connu (aucun module graphite_perfdata n'est configuré pour écrire des données sur ce serveur)

```
graphite_backends      Corse:192.168.1.53,
Reunion:192.168.1.148, Metropole:172.16.0.182
```

Correction : Préciser un serveur qui gère les données du royaume Metropole :

```
graphite_backends      Corse:192.168.1.53,
Reunion:192.168.1.148, Metropole:172.16.0.181
```

```
The server on * is not a known Graphite server
```

? Unknown Attachment

Cas 5 : Le broker gère des royaumes qui ne sont pas configurés dans les graphite_backends de la webui

Vous êtes dans ce cas, si le message suivant apparaît :

```
Realm [*] : No graphics will be displayed as this
realm is not present in "graphite_backends"
parameter
```

Exemple : Le broker broker-metropole-01 gère le royaume Metropole et ses deux sous-royaume Reunion et Corse. Dans la définition de la webui webui-02-validation16-01, le royaume Reunion est manquant. On ne pourra donc pas accéder aux métriques sur les hôtes du royaume Reunion sur la webui webui-02-validation16-01

```
graphite_backends      Corse:192.168.1.53,
Metropole:172.16.0.181
```

Correction : Il faut ajouter le royaume manquant spécifiquement, ou bien mettre le caractère * pour indiquer que tous les royaumes se situent sur le même serveur

```
Realm [*] : No graphics will be displayed as this realm is not
present in "graphite_backends" parameter
```

? Unknown Attachment

```
graphite_backends      Corse:192.168.1.53,  
Reunion:192.168.1.148, Metropole:172.16.0.181
```



Cette erreur n'est présente que si il y a des hôtes dans ce royaume. Si le royaume n'a pas d'hôte, aucune métrique ne sera lues et donc il n'y aura pas d'erreurs.

Cas 6 : Le broker rencontre des erreurs pour lire les données sur les serveurs graphite

Vous êtes dans ce cas, si le message suivant apparaît :

```
Graphite read error happened * times during the  
last 24h, check the /opt/graphite/storage/whisper  
read access on the graphite host
```

? Unknown Attachment

Le broker a rencontré des erreurs lors des dernières 24heures pour lire des métriques.

Correction : Il faut consulter les logs du broker concerné afin de connaître la nature de l'erreur. (problème de droits sur les dossiers de stockage u serveur graphite)