Shinken-healthcheck - Vérifier le bon fonctionnement de Shinken Entreprise

Sommaire

Qu'est-ce que le shinken-healthcheck ?

Le shinken-healthcheck est une commande présente dans toute installation Shinken Entreprise qui permet de vérifier le bon fonctionnement de Shinken Entreprise.

Cet outil est utilisé pour vérifier :

- L'état de l'installation de Shinken Enterprise (*version des démons*).
- L'état des principales options de configuration réseau (*ports, adresses*).
- L'état des modules et sous-modules activés sur les démons.
- L'état des connexions réseau et la synchronisation d'horloge entre les démons.

Le shinken-healthcheck est donc un outil de diagnostic général qui peut détecter les problèmes les plus importants. Cependant, il ne fournit pas autant d'informations et de détail que les checks fournissent par Shinken pour sa propre supervision, par exemple des indicateurs de performance.

Usage

```
shinken-healthcheck
```

Principales options

Option	Option longue	Description
-h	--help	Affiche le message d'aide
-v	--version	Affiche la version de Shinken Entreprise installée
-l	--local	Effectue une vérification des démons locaux seulement
-g	--global	Effectue une vérification complète des démons (*doit être lancé depuis la machine comportant l'Arbiter et le Synchronizer*). Par défaut sur une machine avec un Arbiter et un Synchronizer, un Healthcheck global est effectué sauf si un Healthcheck local est explicitement demandé.
	--debug	Active l'affichage des données de debug dans la sortie de la commande. Cette option est utile seulement dans le cas d'un envoi de ces données aux équipes de support de Shinken Solutions.

-f	--file	Écris la sortie de la commande dans un fichier. La sortie de la commande est également affichée.
	--output-directory	Dossier dans lequel sera placé le fichier de sortie. Par défaut, le dossier courant est utilisé.
	--output-name	Fichier dans lequel sera placé le fichier de sortie. Valeur par défaut : shinken-healthcheck_$(DATE).txt
	--timeout	Temps en secondes à partir duquel un démon sera considéré comme injoignable. Par défaut : 3 secondes
	--show-history	Affiche l'historique des installations et données de Shinken Entreprise sur ce serveur

La commande Shinken-healthcheck sépare sa vérification en plusieurs parties qui sont décrites dans les sections suivantes.

Lancement centralisé de toute l'architecture ou local uniquement

La commande shinken-healthcheck peut être utilisée dans deux modes de fonctionnements différents :

- Vérification globale de toute l'architecture
  - Ce mode est le mode par défaut de la commande
  - La commande vérifie l'ensemble des serveurs à distance et local, telle que définie dans ses fichiers .cfg
  - Peut être lancé avec l'option **--global** ou **-g**
- Vérification locale
  - Ce mode vérifie que les démons locaux à la machine, sans vérifier les notions de royaumes
  - Ce mode est automatiquement sélectionné quand :
    - On est sur le serveur de l'Arbiter spare
    - On est sur un serveur qui n'a pas d'Arbiter
  - Peut être lancé avec l'option **--local** ou **-l**

Informations de version

L'option **--version** ou **-v** donnes les informations suivantes :

- **Original installed version** : C'est la toute première version de Shinken qui a été installée
- **Updated version** : C'est la version actuelle de Shinken
- **Updated patch** : C'est le nom du *patch* actuellement installé. Si aucun patch est installé, alors cette ligne ne s'affiche pas

Vérification de l'architecture

Etat des démons

Le Shinken-healthcheck affiche ensuite pour tous les démons activés dans la configuration, différentes informations indiquant le bon fonctionnement du démon:

- Le type et le nom du démon. Si celui est un Spare, une mention "SPARE" est présente à la suite du nom du démon.
- La configuration est-elle valide ?
- Le démon est-il joignable sur le port trouvé dans la configuration ?
- La version actuelle du démon. S'il y a une différence de version entre l'Arbiter et le démon, un message d'erreur indique cette différence.
- Connexion avec l'Arbiter, ainsi que le décalage de temps entre le démon et l'Arbiter.
- Si plusieurs Arbiters envoient une configuration au démon, un message d'erreur indique qu'un ou plusieurs Arbiters sont en conflit, en précisant l'IP de chacun de ces Arbiters.
- Liste des autres démons à contacter pour pouvoir fonctionner correctement (*section "Talk to"*).
- État des modules et le cas échéant des sous-modules. Si un module a redémarré récemment, un avertissement sera affiché en indiquant la liste des derniers redémarrages du module ou sous-modules.
- Champs spécifiques au démon
  - Liste des tags pour le Poller et Reactionner ainsi que les éventuelles erreurs sur les workers.

```
[ brokers ]
    [broker: broker-master]
        OK:      Configuration seems valid
        OK:      Connection to broker-master is OK at port 7772
        OK:      Daemon version is: [redacted]
        INFO:    This daemon do not have any spare defined
        Arbiters connection:
            Architecture Shinken-lab-validation282:
                OK:      Arbiter arbiter-master: last connection 15s ago
        Modules:
            OK:      Name: Simple-log            Type: simple_log
            OK:      Name: event-manager-writer  Type: event_container
            OK:      Name: WebUI                 Type: webui
            OK:      Name: sla                   Type: sla
            OK:      Name: Livestatus            Type: livestatus
            OK:      Name: Graphite-Perfdata     Type: graphite_perfdata
            Module sla:
                OK:      Connect to MongoDB with address : [172.16.0.191]
            Module WebUI:
                OK:      Connection to WebUI is OK at port 7767
                OK:      Auth_secret is a custom variable
                Submodules:
                    OK:      Name: Mongodb                       Type: mongodb
                    OK:      Name: event-manager-reader          Type: event_container
                    OK:      Name: webui-module-service-weather  Type: webui_module_service_weather
                    OK:      Name: webui-enterprise              Type: webui_enterprise
                    OK:      Name: Cfg_password                  Type: cfg_password_webui
                    OK:      Name: sla                           Type: sla
                    Module Mongodb:
                        OK:      Connect to MongoDB with address : [172.16.0.191]
        Talk to:
            OK:      Reachable scheduler satellite (scheduler-master) at [redacted]
```

Dans l'affichage de l'état des démons, ainsi que dans les sections suivantes, plusieurs états peuvent être retournés :

- **OK** : Tout va bien
- **AT RISK** : Problème pouvant potentiellement nuire au fonctionnement du système.
- **ERROR** : Une erreur bloquante a été détectée.

Cas spécifique au Synchronizer - Vérification du statut du chiffrement des propriétés et données protégées

La sous-section "Encryption status" donne des informations sur le chiffrement des propriétés et données protégées :

- Protection activée ou pas (*et nom de la clé si elle est activée*)
- Si la clé n'a pas été sauvegardée ou si la clé ne peut pas être chargée, une erreur est affichée
- S'il y a une incohérence de configuration du chiffrement (*voir la page :* Mécanismes de sécurisation du chiffrement), le message d'erreur correspondant sera affiché.

```
[synchronizer: synchronizer-master]
    OK:      Connect to MongoDB with address : [172.16.0.191]
    OK:      Auth_secret is a custom variable
    OK:      Connection to daemon is OK at port 7765
    OK:      Connection to Synchronizer UI is OK at port 7766
    OK:      Configuration seems valid
    OK:      Daemon version is: [redacted]
    [Encryption status]
        OK:      Encryption DISABLED
    Modules:
        OK:      Name: Cfg_password                          Type: cfg_password_webui
        OK:      Name: synchronizer-module-database-backup   Type: synchronizer_module_database_backup
    Sources:
        OK:      Name: ip-tag-dmz                            Type: sync-ip-tag
        OK:      Name: sync-regexp-tag                       Type: sync-regexp-tag
```

```
    [Encryption status]
        ERROR:   Unable to load keyfile /etc/shinken/secrets/protected_fields_key : No such file or directory ;you need to restore the key test, using shinken-protected-fiel
keyfile-restore before the synchronizer can run.
```

Royaumes et sous-royaumes

Dans une configuration de Shinken Entreprise, les démons peuvent être répartis sur plusieurs machines.

Dans la commande shinken-healthcheck, les démons sont regroupés en fonction de la machine sur laquelle ils sont installés.

On voit dans l'exemple ci-dessous qu'un Poller est installé sur la machine d'adresse 192.168.1.35, et qu'un Arbiter et un Broker sont installés et activés sur la machine vm3 (*172.16.0.3*).

```
- 192.168.1.35 (192.168.1.35):
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
    [poller: poller-spare] - SPARE
        ERROR:   Cannot contact daemon 192.168.1.35:7771 ( timed out )
        OK:      Configuration seems valid


- vm3 (172.16.0.3):
^^^^^^^^^^^^^^^^^^
    [arbiter: arbiter-vm3]
        OK:      Configuration seems valid
        OK:      Connection to daemon is OK at port 7770
        OK:      Daemon version is: [redacted]
        OK:      This element is defined as the master arbiter
        Modules:
            OK:      Name: synchronizer-import     Type: synchronizer-import
        Talk to:
            ERROR:   Unreachable poller satellite (poller-spare) at http://192.168.1.35:7771 (timed out)
            OK:      Reachable scheduler satellite (scheduler-vm3) at http://vm3:7768
            OK:      Reachable reactionner satellite (reactionner-vm3) at http://vm3:7769
            OK:      Reachable poller satellite (poller-vm3) at http://vm3:7771
            OK:      Reachable broker satellite (broker-vm3) at http://vm3:7772
            OK:      Reachable receiver satellite (receiver-vm3) at http://vm3:7773
    [broker: broker-vm3]
        OK:      Configuration seems valid
```

Si plusieurs royaumes sont définis, la sortie de Shinken-healthcheck organise les machines par royaume et sous-royaumes, puis les démons sont répartis par machine d'installation.

Dans l'exemple ci-contre, on voit que la configuration comporte 4 royaumes, agencés comme suivant :

- Un royaume principal : **France**
  - Un sous Royaume : **Corse**
  - Un sous Royaume : **Sud Ouest**
    - Un sous Royaume : **Bordeaux**

On voit aussi, pour chaque royaume, les démons activés ainsi que la machine sur laquelle ils sont installés. Dans l'exemple de Shinken-healthcheck, on peut faire le récapitulatif suivant :

- Royaume **France** : 8 démons répartis sur 2 machines
  - Machine d'adresse a.a.a.a : Poller
  - Machine master1 (b.b.b.b) : Arbiter, Broker, Poller, Reactionner, Synchronizer, Receiver, Scheduler
- Royaume **France/Corse** : 3 démons installés sur une seule machine
  - Machine master2 (d.d.d.d) : Broker, Poller, Scheduler
- Royaume **France/Sud Ouest** : Un démon installé sur une machine
  - Machine master2 : Broker
- Royaume **France/Sud Ouest/Bordeaux** : 2 démons installés
  - Machine master2 : Poller, Scheduler

Exemple d'architecture

```
    -----------------
    | Realm /France |
    -----------------

        --------------
        | In France/ |
        --------------

            - a.a.a.a (a.a.a.a):
            ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
                [poller: poller-windows1]
                    ....

            - master1 (b.b.b.b):
            ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
                [arbiter: arbiter-master]
                    ...
                [broker: broker-master]
                    ...
                [poller: poller-master1]
                    ...
                [reactionner: reactionner-master]
                    ...
                [synchronizer: synchronizer-master]
                    ...
                [receiver: receiver-1]
```

```
                    ...
                [scheduler: scheduler-master]
                    ...

        ----------------------
        | Realm /France/Corse |
        ----------------------

            -------------
            | In Corse/ |
            -------------

                - master3 (d.d.d.d):
                ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
                    [broker: broker-master-3]
                        ...
                    [poller: poller-master3]
                        ...
                    [scheduler: scheduler-master3]
                        ...

        --------------------------
        | Realm /France/Sud Ouest |
        --------------------------

            -----------------
            | In Sud Ouest/ |
            -----------------

                - master2 (c.c.c.c):
                ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
                    [broker: broker-master-2]
                        ...

            -----------------------------------
            | Realm /France/Sud Ouest/Bordeaux |
            -----------------------------------

                ---------------
                | In Bordeaux/ |
                ---------------

                    - master2 (c.c.c.c):
                    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
                        [poller: poller-master2]
                            ...
                        [scheduler: scheduler-
master2]
                            ...
```

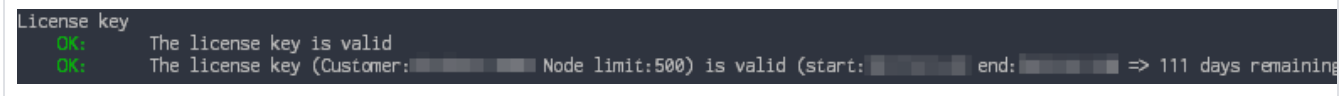
Vérification de la licence

Cette section du Shinken-healthcheck affiche des informations sur la licence en cours.

Elle affiche le propriétaire de la licence, le type de licence, et la date d'expiration de la licence.

Si le nombre de nœuds est dépassé ou que la licence est expirée, une erreur sera affichée dans cette section.

À noter que cette section ne s'affiche que sur les serveurs hébergeant un Arbiter ou un Broker.



Vérification des bibliothèques externes

Shinken Entreprise utilise de nombreuses librairies externes pour fonctionner.

La partie **Local Libraries** du Shinken-healthcheck fournie les informations suivante pour les bibliothèques nécessaires au fonctionnement de Shinken Enterprise :

- Si elles sont bien présentes
- La version de python dans laquelle la bibliothèque est installée
- La version de la bibliothèque
- La liste des dépendances de la bibliothèque et leur version

En cas d'erreur concernant l'une des bibliothèques, un message d'erreur est affiché indiquant la nature de l'erreur.

Vérification des espaces de stockage

Graphite

Shinken Entreprise sauvegarde les données de métrologie dans un serveur Graphite. Si la configuration de la sauvegarde des données de métrologies est simple pour une configuration basique, elle devient rapidement compliquée lorsque la configuration comporte plusieurs royaumes avec plusieurs Brokers.

Lorsque plusieurs Brokers sont dans un même royaume,

- il faut s'assurer que les Brokers n'écrivent pas les données de métrologie au même endroit (*sur le ou les mêmes serveurs Graphite*).
- Il faut également vérifier que toutes les interfaces de Visualisation d'un broker lisent les données de métrologie de tous les royaumes et sous-royaume que le broker gère, afin d'assurer une cohérence des données.
- Aussi, il faut s'assurer de ne pas avoir oublié de configurer au moins un broker d'un royaume pour écrire les données de métrologie dans la base Graphite, sans quoi aucune donnée de métrologie ne sera sauvegardée pour ce royaume.

La commande Shinken-healthcheck possède une section dédiée à la vérification des espaces de stockage des données de métrologie, qui vérifie que les brokers de chaque royaume sont configurés pour sauvegarder les données, que tous les modules Webui sont configurés correctement pour lire les données stockées dans les serveurs graphite de chaque royaume ou sous-royaume et que les serveurs de métrologie sont effectivement joignables et aptes à sauvegarder des données.

Pour cette partie, on utilise une configuration avec un royaume par défaut (*Metropole*) et deux sous royaumes (*Corse et Reunion*).

Vérification de la configuration de la sauvegarde des données de métrologie

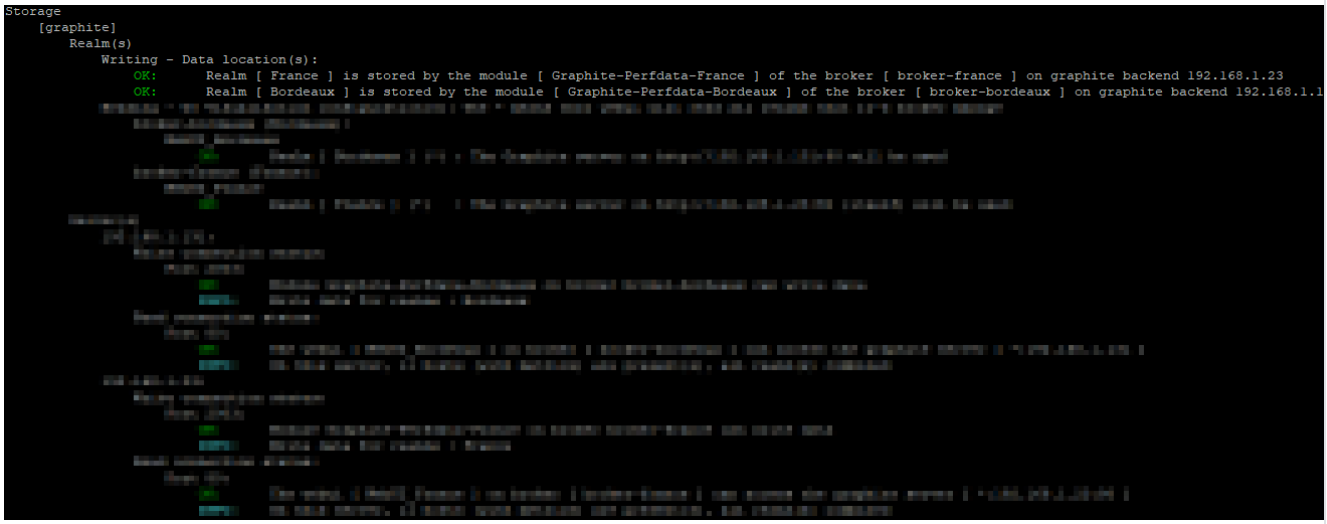
La première section de la vérification vérifie qu'il y a au moins un broker pour chaque royaume ou sous-royaume configuré pour sauvegarder les données de métrologie.

Dans l'exemple ci-contre, on peut voir qu'il y a un broker configuré par royaume, pour sauvegarder les données métrologie sur deux serveurs graphite différent :

- Les données métrologie du royaume **France** sont stockées par le broker **broker-france** sur le serveur graphite **192.168.1.23**
- Les données métrologie du royaume **Bordeaux** sont stockées par le broker **broker-bordeaux** sur le serveur graphite **192.168.1.131**

Si un royaume n'est configuré sur aucun Broker sauvegardant les données de métrologie, une erreur sera affichée.

On note que s'il n'y a aucun hôte à superviser dans le royaume, le Broker n'a pas besoin de sauvegarder de données de métrologie, et donc ne sera pas indiqué en erreur s'il ne sauvegarde pas de données dans Graphite.

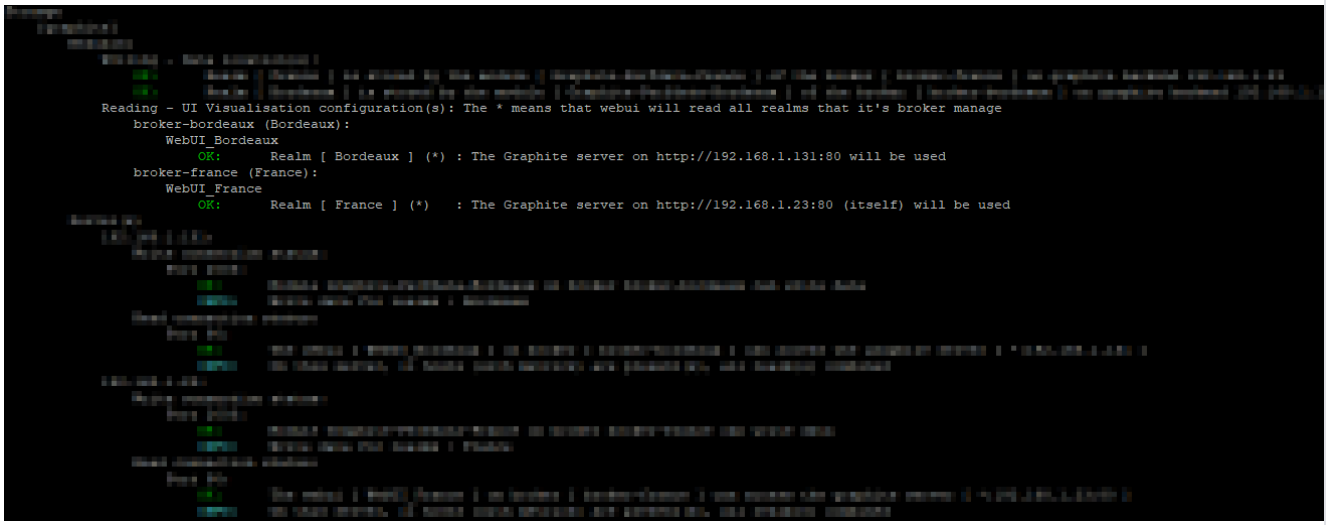


Vérification de la configuration pour la lecture des données de métrologie

Cette section vérifie que chaque module webui a configuré pour chaque royaume (*ou sous royaume*) un serveur graphite que gère le broker.

Dans l'exemple ci-dessous, on peut voir que la vérification de configuration de la lecture s'effectue pour module webui de chaque broker et que pour chaque module, la commande shinken-healthcheck teste la configuration pour les différents royaumes présents sur le broker.

Si un royaume est inconnu ou non géré par le broker, la vérification retournera une erreur pour ce royaume.



Vérification du bon fonctionnement des serveurs graphite

Cette section présente tous les serveurs graphite répertoriés lors de la lecture de la configuration. Un serveur est déterminé par les modules de type "graphite_perfdata" utilisés et également par les graphite_backend de chaque module de type "webui" utilisées.

Chaque serveur graphite est identifié de manière unique par son adresse IP : il est possible que le module "graphite_perfdata" utilise un nom DNS et le module "Webui" une adresse IP. Si les modules utilisent l'adresse de boucle locale (*127.0.0.1*), elle sera aussi remplacée par l'adresse IP du broker sur lequel il tourne.

Pour chaque serveur deux catégories sont affichées : la partie écriture et la partie lecture.

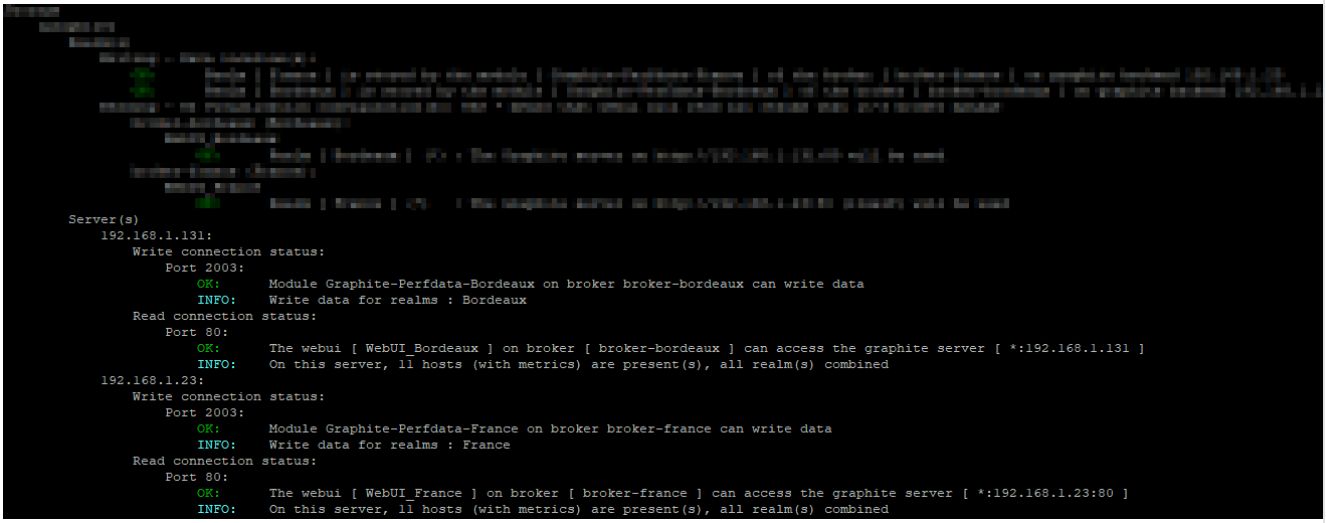
La partie écriture (*Write connexion status*) affiche :

- Le port utilisé pour la communiquer avec le service carbon-cache.
- L'état de la connexion pour chaque module de type "graphite_perdata" qui est configuré pour écrire sur ce serveur.
- Le port utilisé par ce serveur est affiché en information si les connexions des modules sont **OK** (*sinon il sera absent*).

- Les royaumes qui sont écrits sur ce serveur sont affichés en information si les connexions des modules sont **OK** (*sinon il sera absent*).
- Si aucun module n'écrit d'information, cela sera affiché en **AT RISK** .

La partie lecture (*Read connection status*) affiche :

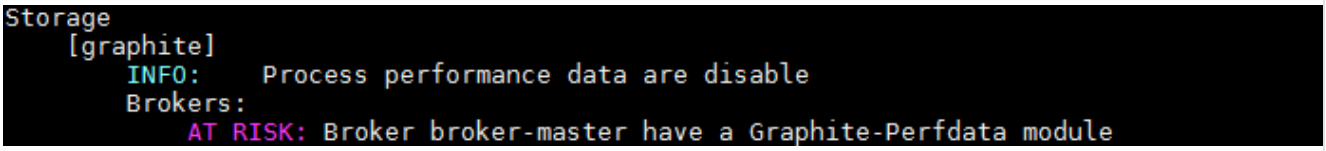
- Le port utilisé pour la communiquer avec le service Graphite.
- L'état de la connexion pour chaque module de type "Webui" qui est configuré pour lire des données sur ce serveur.
- Le nombre d'hôtes qui est accessible sur ce serveur est affiché en information si les connexions des modules sont **OK** (*sinon il sera absent*).
- Si aucun module ne lit d'informations sur ce serveur, cela sera affiché en **AT RISK** .



Affichage lorsque la gestion des données de performance est désactivé

Lorsque dans la configuration avancée de Shinken (*voir la page :* Configuration avancée (shinken.cfg)) l'option de gestion des données de performance (*process_performance_data*) est désactivée, la section "graphite" de la commande Shinken-healthcheck va :

- Informer que la fonctionnalité de gestion des données de performance est désactivée
- Mettre un **AT RISK** pour chaque broker ayant un module "graphite_perfdata" configuré

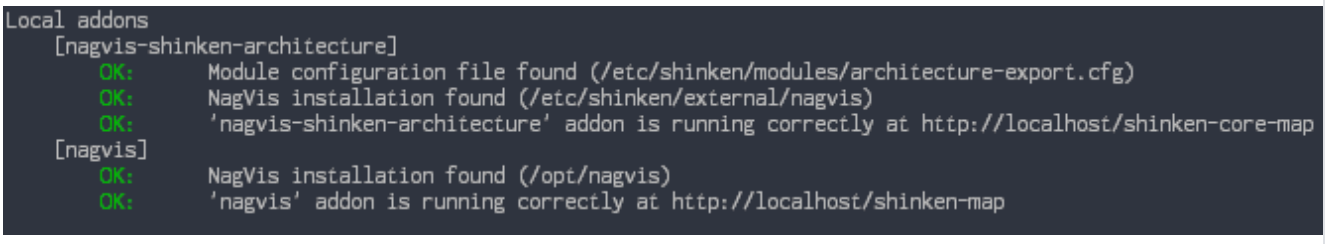


Vérification des addons

La section suivante du Shinken-healthcheck affiche l'état des différents add-ons actifs sur la machine.

La liste des add-ons actuellement activée est affichée, avec pour chacun, leur statut ainsi que l'état de différentes vérifications.

Puisque les add-ons peuvent avoir chacun un fonctionnement différent, les vérifications effectuées différent selon l'add-on.

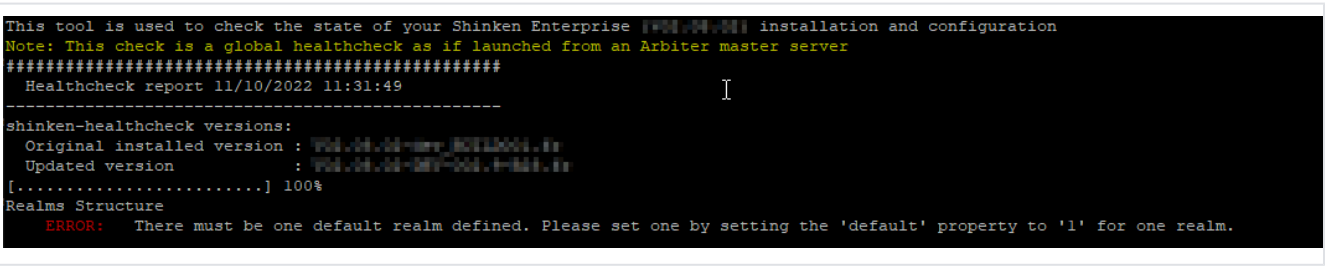


Structure des royaumes

L'avant-dernière section du Shinken-healthcheck l'état de la structure des royaumes.

Si la configuration des royaumes est erronée, les autres vérifications ne sont pas faites et seule cette erreur est affichée.

La copie d'écran montre une erreur, car deux royaumes ont été définis comme royaumes par défaut.



Récapitulatif du healthcheck

Cette dernière section fait résume le nombre d'erreurs, d'informations et de AT RISK présent dans le résultat du Shinken-healthcheck. Cette section permet à l'utilisateur de ne plus faire défiler toutes les lignes du résultat afin de vérifier s'il y a des erreurs ou des avertissements.

Dans notre exemple, on peut savoir aisément qu'il y a 4 erreurs et un avertissement dans le résultat du Shinken-healthcheck.

Historique des installations et de la configuration

Le Shinken-healthcheck affiche par défaut la version initiale d'installation ainsi que la version actuellement installée.

Il est possible d'obtenir des informations supplémentaires sur l'historique de l'installation en utilisant l'option "--show-history" de la commande "shinken-healthcheck":

```
shinken-healthcheck --show-history
```

Ce paramètre affiche des informations sur l'évolution de Shinken Entreprise depuis son installation initiale.

L'utilisation de cette option est surtout pratique pour communiquer avec le support Shinken, qui utilisera les informations remontées pour plus facilement retrouver l'origine du problème.

Historique des mises à jour

L'option --show-history affiche d'abord une liste triée chronologiquement des différentes versions de Shinken installées et désinstallées sur la machine courante.

On peut donc voir facilement, pour une date donnée, quelle était la version de Shinken installée.

Cette option est également utile pour communiquer avec le support Shinken, pour détecter des erreurs de configuration liées à d'anciennes versions de Shinken.

Description des champs de la sortie de l'historique :

Action	Description	Détails
INSTALLATION	Première installation de Shinken.	• **version** : Nom de la version de l'installation d'origine. • **date** : Date d'installation au format *YYYY-MM-DD HH:MM:SS.*
PATCH	Mise à jour de Shinken.	• **on version** : Nom de la version d'origine de Shinken. • **date** : Date d'installation de la mise à jour au format *YYYY-MM-DD HH:MM:SS.* Le nom de la version à la ligne correspond à la version vers laquelle la mise à jour est faite.

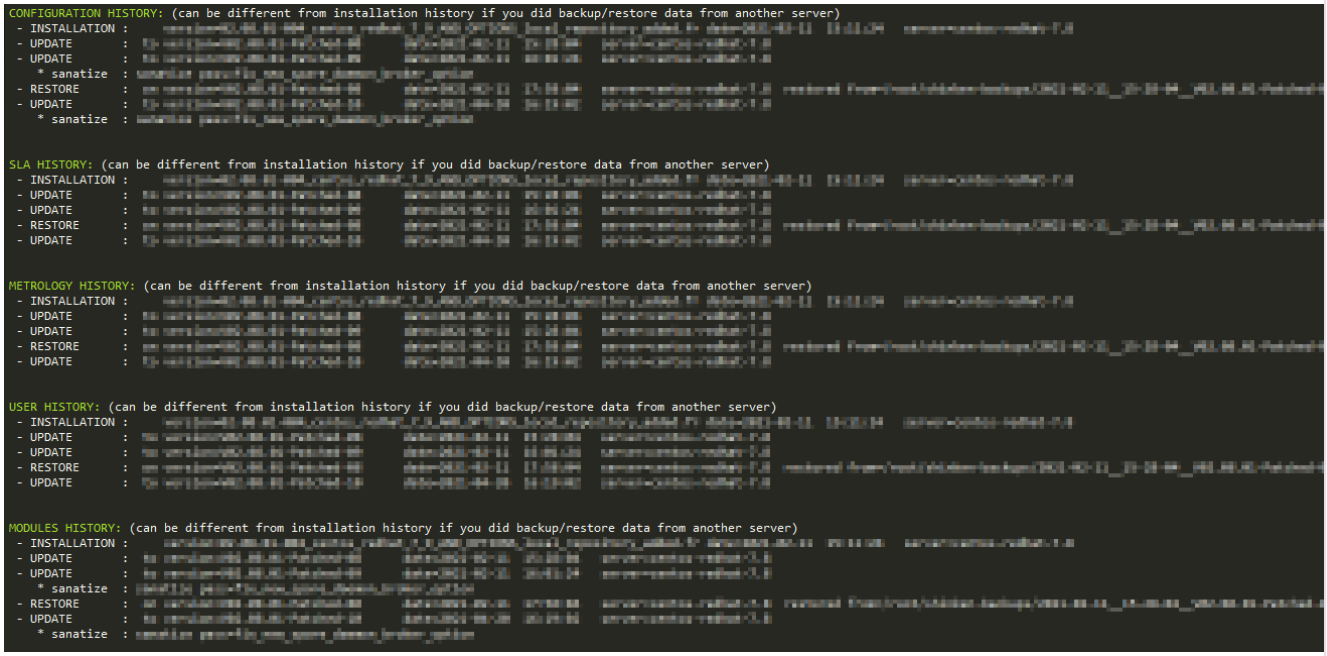
UNPATCH	Désinstalle une mise à jour.	• **on version** : Nom de la version d'origine de Shinken. • **date** : Date de désinstallation de la mise à jour au format *YYYY-MM-DD HH:MM:SS.* Le nom de la version à la ligne correspond à la version du patch désinstallé.

Historique des données

Après l'historique des installations vient l'historique des données de Shinken. Cet historique des données est séparé en plusieurs sections correspondantes chacune à un type de donnée différente :

- **Configuration** : configuration de Shinken, élément de supervision
- **SLA** : historique des états des éléments supervisés et données de SLA
- **Métrologie** : métriques
- **Données** utilisateurs : données sauvegardées dans l'interface de visualisation
- **Modules** : configuration des modules

Action	Description	Détails
INSTALLATION	Première installation de Shinken.	• **version** : Nom de la version de l'installation d'origine. • **date** : Date d'installation au format *YYYY-MM-DD HH:MM:SS.* • **server** : Nom de l'hôte du serveur où a été installé Shinken.
UPDATE	Mise à jour de Shinken.	• **to version** : Nom de la version vers laquelle les données ont été mises à jour. • **date** : Date d'installation de la mise à jour au format *YYYY-MM-DD HH:MM:SS.* • **server** : Nom de l'hôte du serveur où les données se trouvent lors de la mise à jour.
sanatize	Exécution du script de nettoyage des données. Retravaille les données lors d'une mise à jour ou restauration pour les rendre compatibles avec la version installée / restaurée.	• **sanatize pass** : Nom du script exécuté lors de la mise à jour / restauration de version.
RESTORE	Restauration d'une sauvegarde. Le type de données restaurées dépend des options utilisées lors du script de restauration. (*Voir la page :* Shinken-backup et Shinken-restore, les commandes de sauvegarde et de restauration)	• **on version** : Nom de la version vers laquelle les données ont été restaurées. • **date** : Date d'installation de la restauration au format *YYYY-MM-DD HH:MM:SS.* • **server** : Nom de l'hôte du serveur où les données se trouvent lors de la restauration.



Historique des paramètres de chiffrement des données

Le dernier type d'information remonté par l'option --show-history est l'historique des paramètres de chiffrement des données sensibles.

À chaque fois que les paramètres de chiffrement sont modifiés et déclenchent un changement dans le chiffrement des données (*activation, désactivation, changement des champs chiffrés*), une entrée sera ajoutée dans la liste des modifications.

L'état d'activation, le nom de la clé utilisée, l'état de sauvegarde de la clé ainsi que la liste des définitions des données à chiffrer sont affichés pour chaque entrée de la liste.

Notez que le statut de sauvegarde de la clé peut être différent du statut affiché dans la section **[Encryption Status]** :

- Dans la section **[Encryption Status]**, Shinken-healthcheck affiche l'état actuel.
- Dans l'historique, Shinken-healthcheck affiche l'état au moment de la modification de la configuration.

Seuls les cinq derniers changements sont conservés.