Hardware Health Switch SNMPv1v2 - Switch-SNMPv1v2-detailed

Sommaire

Les métriques dans Shinken

Les vérifications faites lors de la supervision des équipements peuvent fournir des mesures en plus de l'état. Ces mesures (*ou donnée de performance, métriques*) peuvent être de tout type.

Par exemple :

- Un check "Memory" sur une machine Linux pourra donner la quantité de mémoire utilisée, de mémoire libre et de mémoire totale.
- Un check sur un switch pourra donner les statistiques de transfert des différentes interfaces réseau.
- Un check sur une application pourra donner le nombre d'utilisateurs actuellement sur l'application, le nombre de nouveaux utilisateurs sur la journée, etc...

Ces mesures sont fournies selon un format définit par le format de sortie des sondes (*voir la page* Les Sondes).

À chaque vérification d'état d'un élément, le module Graphite-Perfdata (*module de Broker*) enregistre les mesures dans la base de données Graphite.

Graphite est une base de données "time series" ce qui va permettre d'associer une date à chaque mesure et de permettre la consultation de ces données sous forme de courbe dans l'interface de Shinken.

Stockage des métriques par Shinken

Shinken Entreprise utilise la base de données Graphite (https://graphiteapp.org/) pour stocker les métriques.

Cette base est constituée de deux éléments :

- carbon-cache : le démon qui gère l'écriture et le stockage des données, écoute par défaut sur le port 2003.
- graphite (*graphite-web*) : le module d'Apache qui permet de lire les données, écoute par défaut sur le port 80.

Dans Shinken, c'est le Broker qui interagit avec Graphite par le biais de deux modules :

- Le module de type `"graphite_perfdata"` permet de sauvegarder les métriques (*voir la page* Module Graphite-Perfdata)
- Le module de type "webui" permet de visualiser les métriques.

Le répertoire de stockage des données de graphite est "/opt/graphite/storage/whisper" par défaut.

Il est conseillé de superviser l'utilisation du disque qui contient la partition où sont les données de Graphite. Comme chaque check de chaque hôte peut générer des métriques, le volume de données écrites peut rapidement devenir important. Il peut donc être judicieux de dédier un disque spécifiquement à ces données et d'opter pour un support offrant un bon débit d'écriture, comme un SSD.

La version du Graphite livré est 1.1.8.

Consultation des métriques

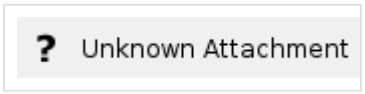
Depuis l'Interface de Visualisation

Il est possible de consulter les métriques depuis deux pages de l'Interface de Visualisation :

- Dans les tableaux de bord, via le widget graphique (*voir la page* Widget Graphique).
- Dans le détail d'un hôte/cluster via l'onglet graphique (*voir la page* Onglet Graphiques).

Lorsqu'un utilisateur veut consulter une métrique dans l'Interface de Visualisation, l'interface requête Graphite via Apache pour récupérer la métrique demandée.

Dans le cas d'une architecture complexe avec plusieurs royaumes, il peut y avoir plusieurs serveurs de stockage des métriques. Dans ce cas, l'Interface de Visualisation trouve automatiquement le serveur Graphite à interroger pour renvoyer les métriques demandées.



Depuis des outils externes

Des outils externes à Shinken peuvent également être utilisés pour visualiser les métriques :

- **via l'interface de Graphite**, accessible par défaut sur le port 80 ;
- **ou à l'aide de l'outil externe Grafana** (*voir la page* Grafana - v8.3.2).

En cas d'utilisation d'outils externes (*par exemple Grafana*) pour consulter les métriques, il faut également :

- configurer la récupération des données de l'inventaire sur les serveurs carbon-cache,
- autoriser les connexions au serveur d'inventaire depuis les serveurs carbon-cache.

Voir la section Correspondance UUID Nom de l'élément pour plus d'information.

Configuration de graphite

Que la visualisation des métriques se fasse via l'interface de Shinken ou un outil externe, il peut être nécessaire d'adapter la configuration de Graphite. Il est possible :

- d'autoriser les connexions externes (*non localhost*),
- de changer le port de Graphite,
- activer le chiffrement HTTPS dans Graphite,
- configurer la Correspondance UUID Nom de l'élément pour les outils externes (*par exemple Grafana*).

Autoriser les connexions externes (non localhost) à Graphite

Par défaut, et par mesure de sécurité, Graphite n'écoute que sur l'interface locale (*127.0.0.1*). Toute requête provenant d'un serveur externe est donc refusée.

Si la base Graphite se trouve sur une machine différente du Broker, alors il faut que la base accepte les connexions externes pour sauvegarder et renvoyer les métriques.

RHEL / CentOS 7 ou RHEL / Alma / Rocky 8

Pour autoriser des connexions externes à accéder à Graphite, il faut modifier la configuration d'Apache de Graphite **/etc/httpd/conf.d/graphite.conf** :

```
<VirtualHost 127.0.0.1:80>
```

à remplacer par

```
<VirtualHost IP_INTERFACE:80>
```

avec :

- **IP_INTERFACE :** l'adresse de l'interface sur laquelle faire l'écoute. Par défaut, l'écoute n'est faite que sur l'interface locale (*127.0.0.1*). Utiliser "*" pour écouter sur toutes les interfaces

Pour que les changements soient pris en compte, il faut redémarrer le service d'Apache (*httpd*).

```
systemctl restart httpd
```

Debian 13

Pour autoriser des connexions externes à accéder à Graphite, il faut modifier la configuration d'Apache de Graphite **/etc/apache2/sites-available /graphite.conf** :

```
<VirtualHost 127.0.0.1:80>
```

à remplacer par

```
<VirtualHost IP_INTERFACE:80>
```

avec :

- **IP_INTERFACE :** l'adresse de l'interface sur laquelle faire l'écoute. Par défaut, l'écoute n'est faite que sur l'interface locale (*127.0.0.1*). Utiliser "*" pour écouter sur toutes les interfaces

Pour que les changements soient pris en compte, il faut redémarrer le service d'Apache (*apache2*).

```
systemctl restart apache2
```

Autoriser le trafic réseau externe vers le démon carbon-cache dans le pare-feu

Lorsque la base Graphite **n'est pas** sur le même serveur que le module Graphite-Perfdata (*module de Broker*), il faut aussi autoriser le flux réseau sur le port de carbon-cache **sur le serveur de Graphite**.

Exemple de commandes pour ajouter le port si le firewall utilisé est **firewalld :**

```
firewall-cmd --add-port=2003/tcp
firewall-cmd --runtime-to-permanent
```

Changer le port de Graphite

Pour changer le port de graphite, il faut le modifier dans :

- Le fichier de configuration de Graphite
- Le fichier de configuration d'Apache

Changer le port dans la configuration de Graphite

RHEL / CentOS 7 ou RHEL / Alma / Rocky 8

Par défaut, Graphite est configuré pour accepter les connexions sur le port 80. Dans son fichier de configuration **/etc/httpd/conf.d/graphite.conf** :

/etc/httpd/conf.d/graphite.conf

```
[ ... ]
<VirtualHost 127.0.0.1:80>
[ ... ]
```

Par exemple pour écouter sur le port 8080 :

/etc/httpd/conf.d/graphite.conf

```
[ ... ]
<VirtualHost 127.0.0.1:8080>
[ ... ]
```

Debian 13

Par défaut, Graphite est configuré pour accepter les connexions sur le port 80. Dans son fichier de configuration **/etc/apache2/sites-available/graphite.conf** :

/etc/apache2/sites-available/graphite.conf

```
[ ... ]
<VirtualHost 127.0.0.1:80>
[ ... ]
```

Par exemple pour écouter sur le port 8080 :

/etc/apache2/sites-available/graphite.conf

```
[ ... ]
<VirtualHost 127.0.0.1:8080>
[ ... ]
```

Changer le port dans la configuration d'Apache

RHEL / CentOS 7 ou RHEL / Alma / Rocky 8

Par défaut, Apache écoute uniquement sur le port 80, pour changer le port par défaut ou en ajouter d'autres, il faut aller modifier le fichier **/etc/httpd/conf/httpd.conf**.

Dans ce fichier, remplacer la partie où est écrit "Listen 80" avec le port souhaité :

/etc/httpd/conf/httpd.conf

```
[ ... ]
Listen 80
[ ... ]
```

Par exemple pour écouter sur le port 8080 :

/etc/httpd/conf/httpd.conf

```
[ ... ]
Listen 8080
[ ... ]
```

Il est aussi possible d'ouvrir plusieurs ports dans ce fichier, mais seul celui défini dans le VirtualHost sera accessible depuis l'extérieur du serveur.

/etc/httpd/conf/httpd.conf

```
[ ... ]
Listen 80
Listen 8080
[ ... ]
```

Pour que les changements soient pris en compte, il faut redémarrer le service d'Apache (*httpd*).

```
systemctl restart httpd
```

Debian 13

Par défaut, Apache écoute uniquement sur le port 80, pour changer le port par défaut ou en ajouter d'autres, il faut aller modifier le fichier **/etc/apache2/ports.conf**.

Dans ce fichier, remplacer la partie où est écrit "Listen 80" avec le port souhaité :

/etc/apache2/ports.conf

```
[ ... ]
Listen 80
[ ... ]
```

Par exemple pour écouter sur le port 8080 :

/etc/apache2/ports.conf

```
[ ... ]
Listen 8080
[ ... ]
```

Il est aussi possible d'ouvrir plusieurs ports dans ce fichier, mais seul celui défini dans le VirtualHost sera accessible depuis l'extérieur du serveur.

/etc/apache2/ports.conf

```
[ ... ]
Listen 80
Listen 8080
[ ... ]
```

Pour que les changements soient pris en compte, il faut redémarrer le service d'Apache (*apache2*).

```
systemctl restart apache2
```

Configurer Shinken pour utiliser le nouveau port de Graphite

Sur la machine de l'Arbiter, il faut configurer les différents modules de WebUI et Graphite-Perfdata qui se connectent à Graphite pour utiliser le nouveau port.

- Dans le fichier de la configuration du module de WebUI **/etc/shinken/modules/webui.cfg** modifier le paramètre :

/etc/shinken/modules/webui.cfg

```
[ ... ]

graphite_backends                                   *=http://ADRESS_SERVER_GRAPHITE:PORT

[ ... ]
```

avec **PORT** le nouveau port

- Dans le fichier de la configuration du module Graphite-Perfdata, modifier l'URL d'envoi de l'inventaire, si nécessaire (*c'est-à-dire, si des outils externes comme Grafana sont utilisés pour consulter les métriques*) :

/etc/shinken/modules/graphite.cfg

```
[ ... ]

  broker__module_graphite_perfdata__inventory_push__url                                  http://ADRES
S_SERVER_GRAPHITE:PORT/migrate

[ ... ]
```

avec **PORT** le nouveau port

Il faut ensuite redémarrer l'Arbiter.

```
service-shinken-arbiter restart
```

Dans le cas d'un cluster Graphite, l'adresse du serveur à mettre correspond à celle de la machine où se trouve le carbon-relay.

Autoriser le trafic réseau du nouveau port dans le pare-feu

Si la base Graphite **n'est pas** sur le même serveur que le Broker, il faut aussi autoriser le flux réseau vers le nouveau port de Graphite dans le pare-feu.

Exemple de commandes pour ajouter le port 8080 si le firewall utilisé est **firewalld :**

```
firewall-cmd --add-port=8080/tcp
firewall-cmd --runtime-to-permanent
```

Erreur lors du démarrage d'Apache

Après avoir changé les ports dans les fichiers de configuration et redémarrer Apache, il est possible d'avoir une erreur du type "Permission denied".

Il est possible que ce soit SELinux qui bloque le port choisi. Il faut alors configurer SELinux pour qu'il autorise le nouveau port.

Pour connaitre les ports httpd autorisés par SELinux :

```
semanage port -l | grep http
```

Pour ajouter un port pour la règle *http_port_t,* lancer la commande :

```
semanage port -a -t http_port_t -p tcp NOVEAU_PORT
```

> Modifier les règles SELinux, c'est **étendre les permissions**, donc :
>
> - Ouvrir potentiellement une surface d'attaque,
> - Risquer de compromettre l'isolation qu'assure SELinux entre services,
>
> Il peut être préférable d'utiliser un port déjà autorisé plutôt que d'en ajouter un.

Pour plus d'informations, voir la documentation de SELinux : https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/security-enhanced_linux/sect-security-enhanced_linux-top_three_causes_of_problems-how_are_confined_services_running

Activation de HTTPS pour Graphite

Par défaut, Graphite utilise le protocole HTTP.

Pour activer le protocole HTTPS, il faut :

- ouvrir le port 443 d'Apache s'il n'est pas ouvert,
- activer le HTTPS dans Graphite,
- configurer Shinken pour prendre en compte le changement

Ouvrir le port 443 d'Apache

RHEL / CentOS 7 ou RHEL / Alma / Rocky 8

Pour vérifier si Apache écoute sur le port 443 :

```
netstat -laputen | grep 443
```

Exemple de retour :

Apache accepte les connexions HTTPS

```
netstat -laputen | grep 443
tcp6       0      0 :::443                  :::*                    LISTEN      0          114741     15195
/httpd
```

Si le port 443 n'est pas ouvert sur Apache, il faut ajouter la ligne suivante dans le fichier **/etc/httpd/conf.d/ssl.conf** :

```
[ ... ]
Listen 443 https
[ ... ]
```

Pour que les changements soient pris en compte, il faut redémarrer le service d'Apache :

```
systemctl restart httpd
```

Debian 13

Pour vérifier si Apache écoute sur le port 443 :

```
netstat -laputen | grep 443
```

Exemple de retour :

Apache accepte les connexions HTTPS

```
netstat -laputen | grep 443
tcp6       0      0 :::443                  :::*                    LISTEN      0          114741     15195
/apache2
```

Si le port 443 n'est pas ouvert sur Apache, lancer les commandes suivantes pour activer le SSL :

```
a2enmod ssl
systemctl restart apache2
```

Activer le HTTPS de Graphite

RHEL / CentOS 7 ou RHEL / Alma / Rocky 8

Pour activer le HTTPS dans Graphite, il faut modifier la partie "VirtualHost" du fichier **/etc/httpd/conf.d/graphite.conf**.

Par défaut, le VirtualHost de ce fichier ressemblera à ça :

/etc/httpd/conf.d/graphite.conf

```
[ ... ]
<VirtualHost 127.0.0.1:80>
               ServerName graphite
        DocumentRoot "/opt/graphite/webapp"
        ErrorLog /var/log/graphite/graphite-webapp.error.log
        CustomLog /var/log/graphite/graphite-webapp.access.log common

 [ ... ]
```

Pour activer HTTPS, il faut ajouter dans le fichier les trois champs SSLEngine, SSLCertificateFile et SSLCertificateKeyFile et changer le port (*443*) :

/etc/httpd/conf.d/graphite.conf

```
[ ... ]
<VirtualHost 127.0.0.1:443>
               ServerName graphite
        DocumentRoot "/opt/graphite/webapp"
        ErrorLog /var/log/graphite/graphite-webapp.error.log
        CustomLog /var/log/graphite/graphite-webapp.access.log common

            SSLEngine on
        SSLCertificateFile "/path/to/www.example.com.cert"
        SSLCertificateKeyFile "/path/to/www.example.com.key"
[ ... ]
```

Bien préciser le chemin vers les certificats utilisés pour les champs SSLCertificateFile et SSLCertificateKeyFile.

Pour que les changements soient pris en compte, il faut redémarrer le service d'Apache :

```
systemctl restart httpd
```

Debian 13

Pour activer le HTTPS dans Graphite, il faut modifier la partie "VirtualHost" du fichier **/etc/apache2/sites-available/graphite.conf.**

Par défaut, le VirtualHost de ce fichier ressemblera à ça :

/etc/apache2/sites-available/graphite.conf

```
[ ... ]
<VirtualHost 127.0.0.1:80>
                ServerName graphite
        DocumentRoot "/opt/graphite/webapp"
        ErrorLog /var/log/graphite/graphite-webapp.error.log
        CustomLog /var/log/graphite/graphite-webapp.access.log common

 [ ... ]
```

Pour activer HTTPS, il faut ajouter dans le fichier les trois champs SSLEngine, SSLCertificateFile et SSLCertificateKeyFile et changer le port (*443*) :

/etc/apache2/sites-available/graphite.conf

```
[ ... ]
<VirtualHost 127.0.0.1:443>
                ServerName graphite
        DocumentRoot "/opt/graphite/webapp"
        ErrorLog /var/log/graphite/graphite-webapp.error.log
        CustomLog /var/log/graphite/graphite-webapp.access.log common

            SSLEngine on
        SSLCertificateFile "/path/to/www.example.com.cert"
        SSLCertificateKeyFile "/path/to/www.example.com.key"
[ ... ]
```

Bien préciser le chemin vers les certificats utilisés pour les champs SSLCertificateFile et SSLCertificateKeyFile.

Pour que les changements soient pris en compte, il faut redémarrer le service d'Apache :

```
systemctl restart apache2
```

Configurer Shinken pour utiliser SSL avec Graphite

Sur la machine de l'Arbiter, il faut configurer les différents modules de WebUI et Graphite-Perfdata qui se connectent à Graphite pour utiliser le protocole HTTPS.

- Dans le fichier de la configuration du module de WebUI **/etc/shinken/modules/webui.cfg** modifier le paramètre :

/etc/shinken/modules/webui.cfg

```
[ ... ]

graphite_backends                                *=https://ADRESS_SERVER_GRAPHITE:443

[ ... ]
```

- Dans le fichier de la configuration du module Graphite-Perfdata, modifier l'URL d'envoi de l'inventaire, si nécessaire (*c'est-à-dire, si des outils externes comme Grafana sont utilisés pour consulter les métriques*)

/etc/shinken/modules/graphite.cfg

```
[ ... ]

  broker__module_graphite_perfdata__inventory_push__url                                https://ADRESS_SERV
ER_GRAPHITE:443/migrate

[ ... ]
```

Il faudra ensuite redémarrer l'Arbiter.

```
service-shinken-arbiter restart
```

Dans le cas d'un cluster Graphite, l'adresse du serveur à mettre correspond à celle de la machine où se trouve le carbon-relay.

Supervision d'un Graphite en HTTPS par Shinken

Shinken fournit le modèle d'hôte shinken-graphite pour la supervision de l'état de la base Graphite. Le check ***Shinken Graphite Status*** livré dans ce modèle interroge l'état de la base en utilisant l'adresse de l'hôte Graphite (*Clé d'import : address*).

Il est donc important de s'assurer que le certificat SSL/TLS de Graphite est émis pour correspondre à cette adresse.

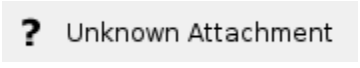
Par exemple, si l'adresse de l'hôte dans Shinken est 'localhost' mais que le certificat est émis pour '127.0.0.1, le check de supervision échouera.

Correspondance UUID Nom de l'élément

Shinken utilise l'UUID de l'élément (*hôte/cluster/check*) pour l'identification des métriques. Cette identification par un ID unique permet de conserver les métriques lors d'un renommage de l'élément.

- Mais les outils externes accédant à Graphite (*par exemple Grafana*) ne sont pas tous capables de comprendre la correspondance NOMUUID.
- Pour résoudre ce problème, Shinken a mis une passerelle pour les outils externes.
  - Par défaut, les appels à Graphite renvoient les noms comme clef des métriques pour les outils externes.
  - Le Broker et ses modules interrogent Graphite avec un paramètre additionnel qui permet d'accéder aux métriques via les UUID.

Graphite a besoin de mettre à jour sa table de correspondance des noms pour les nouveaux éléments et ceux qui ont été renommés.



- Cette correspondance est fournie par le serveur d'inventaire associé aux modules de métrologie de Shinken.
  - Graphite obtient les paramètres d'accès à ces serveurs via le fichier **/opt/graphite/conf/shinken_inventory.conf**
- Cette recherche n'est faite que si une requête par nom est demandée à Graphite et que la table n'est plus à jour.

- Afin de gérer le cas où des hôtes sont renommés vers de noms d'hôte qui existaient précédemment, Graphite vide son cache lors d'une nouvelle mise en production
    - afin que tous les processus de Graphite/Apache soient mis au courant, le fichier */opt/graphite/storage/whisper/.cacheinvalidation* est mis à jour
        - ce fichier ne doit pas être modifié
        - en cas de problème, il est recréé, et le cache vidé

Paramètres de connexion aux serveurs d'inventaire

- Graphite se base sur les informations du fichier **/opt/graphite/conf/shinken_inventory.conf** pour aller chercher les informations qui lui permettront d'assurer la correspondance entre les noms et les ID

Nom	Type	Unité	Défaut	Description
`ENABLE`	Booléen	---	**1**	Permets d'activer ou désactiver la recherche des correspondances entre les ID et les noms (*1 pour activer, 0 pour désactiver*).
`URI`	Liste d'URI	---	**http://localhost:52000/inventory/**	URL séparées par des virgules. • Permet de contacter chacun des modules de métrologie qui fournit des métriques à ce serveur Graphite. • Si le serveur d'inventaire utilise SSL, il faudra utiliser https au lieu de http. **Exemple :** https://ip-broker01:52000/inventory/**,**https://ip-broker02:52000/inventory/**,**https://ip-broker03:52000/inventory/
`TIMEOUT`	Numérique	---	**10**	Timeout général, utilisé pour les opérations bloquantes comme les tentatives de connexion à un serveur d'inventaire, par exemple. (*secondes*).

RHEL / CentOS 7 ou RHEL / Alma / Rocky 8

Après chaque modification du fichier de configuration, un redémarrage du service Apache est nécessaire afin que Graphite prenne en compte les changements.

```
systemctl restart httpd
```

Debian 13

Après chaque modification du fichier de configuration, un redémarrage du service Apache est nécessaire afin que Graphite prenne en compte les changements.

```
systemctl restart apache2
```

Autoriser les connexions aux serveurs d'inventaire

Configurer les modules de métrologie Graphite

Si le serveur Graphite et les Brokers avec les modules de métrologie Graphite sont sur des machines différentes, il faut configurer le serveur d'inventaire des modules de métrologie Graphite pour écouter sur les IP publiques de leur machine,

Pour cela, sur le serveur de **l'Arbiter**, il faut éditer les fichiers de configuration des modules Graphite **/etc/shinken/modules/graphite.cfg** :

- Décommenter et changer l'interface **127.0.0.1** en **0.0.0.0**

/etc/shinken/modules/graphite.cfg

```
broker__module_graphite_perfdata__inventory_server__address  0.0.0.0
```

- Décommenter et préciser l'IP de Graphite où le module enverra les modifications de l'inventaire :

/etc/shinken/modules/graphite.cfg

```
broker__module_graphite_perfdata__inventory_push__url  http://IP_GRAPHITE/migrate
```

- Il faut redémarrer l'Arbiter pour appliquer le changement de configuration :

```
service-shinken-arbiter restart
```

Ouvrir le port du serveur d'inventaire sur le pare-feu

Sur les machines des Brokers **avec un serveur d'inventaire**, il faut ouvrir sur le pare-feu les connexions sur le port de l'inventaire (*52000 par défaut*). On utilisera le pare-feu **firewalld** comme exemple.

Pour lister les ports autorisés sur une machine :

```
firewall-cmd --list-ports
```

Exemple de retour et interprétation :

```
80/tcp 7763/tcp 7765/tcp 7766/tcp 7767/tcp 7768/tcp 7769/tcp 7770/tcp 7771/tcp 7772/tcp 7773/tcp 7777/tcp
7780/tcp 50000/tcp
```

Le port 52000/tcp (*port par défaut du serveur d'inventaire du module de métrologie Graphite*), n'est pas listé, il est donc bloqué.

Pour autoriser les connexions sur le port 52000 :

```
firewall-cmd --add-port=52000/tcp
firewall-cmd --runtime-to-permanent
```

Compatibilité historique

En cas d'impossibilité d'accès au serveur d'inventaire des modules de métrologie (*ports bloqués, paramètres par défaut incompatibles avec la configuration…,*), Graphite peut utiliser l'ancienne méthode que Shinken avait déployé pour fournir ces informations avec MongoDB.

- L'accès est configuré dans Graphite dans le fichier **/opt/graphite/conf/mongodb.conf**.

L'accès via Mongo est déprécié et est voué à disparaître.

En effet, Graphite ne peut consulter qu'une seule base Mongo pour obtenir les correspondances de noms, il est ainsi obligé d'utiliser la base centrale, qui est souvent aussi la plus chargée

Configuration de l'accès à MongoDB

Pour se connecter au serveur Mongo, deux méthodes sont disponibles :

- **Connexion directe :** Par défaut, mais non sécurisée.
- **Tunnel SSH :** Shinken se connecte au serveur Mongo au travers d'un module SSH pour plus de sécurité

Connexion directe au serveur Mongo

Par défaut, Graphite se connecte de manière directe au serveur Mongo pour y lire et écrire sa table de correspondance.

Dans la configuration de Graphite, on sait que la connexion se fait de manière directe lorsque le paramètre "USE_SSH_TUNNEL" est à 0.

Cette méthode de connexion a pour avantage d'être facile à configurer au niveau de Shinken. Par contre, elle oblige à permettre l'accès à la base Mongo au monde extérieur, et donc s'exposer à des problèmes de sécurité.

- La sécurisation de la base Mongo est bien sûr toujours possible (*voir la page Sécurisation des connexions aux bases MongoDB*) mais bien plus complexe à mettre en place.

- La méthode de connexion par SSH est ainsi préférable pour des raisons pratiques et de sécurité.

Connexion par SSH au serveur Mongo

Graphite peut également se connecter au serveur mongo par tunnel SSH (*pour des raisons de sécurité*).

- En effet, le paramétrage de MongoDB (*/etc/mongod.conf*) permet de définir sur quelle adresse ce dernier écoute les requêtes.
  - En n'autorisant seulement l'adresse 127.0.0.1, cela évite d'ouvrir la base au monde extérieur.
    - Dans la configuration du serveur MongoDB (*/etc/mongod.conf*), s'assurer que le paramètre "*bind_ip* " est positionné pour n'écouter que sur l'interface locale :
    - `bind_ip= 127.0 . 0.1`

- Pour paramétrer la connexion à MongoDB depuis Graphite, il faut éditer les options suivantes (*dans* ***/opt/graphite/conf/mongodb.conf***)

Nom	Type	Unité	Défaut	Description
`URI`	Texte	---	**mongodb://ADRESSE-SERVEUR-MONGO/?w=1&fsync=false**	URI du serveur Mongo L'adresse de la base Mongo à utiliser est celle configurée dans le module Graphite-Perfdata (voir la page Module Graphite-Perfdata).
`DATABASE`	Texte	---	**shinken**	Nom de la base contenant les données d'inventaire sur le serveur Mongo
`COLLECTION`	Texte	---	**metrology_inventory**	Nom de la collection contenant les données d'inventaire
`USE_SSH_TUNNEL`	Booléen	---	**0**	Activer la connexion à Mongo par Tunnel SSH
`SSH_USER`	Texte	---	**shinken**	Utilisateur sur le serveur Mongo à contacter pour établir la connexion
`SSH_KEYFILE`	Texte	---	**/opt/graphite/conf/id_rsa**	Doit pointer vers la clé ssh privée sur le serveur Shinken. **Attention** : Apache n'ayant pas les droits d'accès au répertoire ~shinken, il faut copier la clé dans **/opt/graphite/conf/id_rsa** et la rendre accessible par l'utilisateur **apache** (*chown apache:apache /opt/graphite/conf/id_rsa*)
`SSH_TUNNEL_TIMEOUT`	Entier	---	**5**	Timeout utilisé pour tester le tunnel SSH avant de lancer la connexion mongo

Après tous changements du fichier de configuration, il faut redémarrer Apache pour que Graphite prenne les modifications en compte

RHEL / CentOS 7 ou RHEL / Alma / Rocky 8

```
systemctl restart httpd
```

Debian 13

```
systemctl restart apache2
```

Graphite étant hébergé par le service apache, il n'a pas accès au répertoire **/var/lib/shinken** et il n'a donc pas accès à la clé SSH **/var/lib/shinken/.ssh/id_rsa**.

C'est pour cette raison que la clé SSH utilisée pour le tunnel est situé dans **/opt/graphite/conf/id_rsa**.

Deux solutions sont disponibles :

- Générer une nouvelle clé SSH pour apache / graphite (*voir la page* Création automatique et gestion de la clé SSH de l'utilisateur shinken)
  - Lors de la génération de la clé, il est possible de spécifier directement le chemin suivant : **/opt/graphite/conf/id_rsa**
  - Il faudra ajouter cette nouvelle clé publique (*/opt/graphite/conf/id_rsa.pub*) sur le/les serveurs MongoDB (*dans le fichier ~shinken/.ssh/authorized_keys*)
  - Cette clé sera indépendante et non impactée par un changement de clé SSH sur l'utilisateur "shinken".

- Utiliser la clé SSH de l'utilisateur "shinken" présent sur le serveur.
  - La clé publique est sûrement déjà présente sur les serveurs MongoDB.
  - Il faut copier la clé privée et changer les droits pour l'utiliser et la maintenir à jour en cas de changement.

```
cp /var/lib/shinken/.ssh/id_rsa* /opt/graphite/conf/
chown apache:apache /opt/graphite/conf/id_rsa
```

Attention : un lien symbolique entre les deux fichiers ne fonctionnera pas, car l'utilisateur apache n'a pas les droits suffisants pour lire le fichier originel, et SSH refusera d'utiliser une clé dont les droits d'accès sont trop permissifs.

Vérification du bon fonctionnement de graphite

Droits d'accès aux métriques

Pour la lecture des métriques, Graphite se base sur Apache pour fournir un service Web facilement utilisable par d'autres logiciels.

- Pour avoir le droit de lire les métriques, il faut alors que le dossier de stockage des métriques **/opt/graphite/storage/whisper** et ses fils soient possédés par l'utilisateur et le groupe Apache (*apache:apache*).
- Lors de manipulation manuelle sur ces emplacements disques parfois volumineux, il arrive que les droits de **/opt/graphite/storage/whisper** soient modifiés par le système, ce qui empêche la lecture des métriques par Graphite et par conséquent par Shinken (*permission refusée par le système*).

Les commandes suivantes permettent de rétablir les droits nécessaires :

```
chmod -R 0755 /opt/graphite/storage/ /var/log/graphite
chown -R apache:apache /opt/graphite/storage/ /var/log/graphite
```

Vérification de carbon-cache, le demon écrivain

Pour vérifier que le démon **carbon-cache** fonctionne, la première vérification est l'existence de son processus :

```
$ ps axjf | grep carbon-cache
1 21989 21988 21988 ? -1 Sl 48 1202:07 /usr/bin/python /opt/graphite/bin/carbon-cache.py start --config=/opt
/graphite/conf/carbon.conf --pidfile=/opt/graphite/storage/carbon-cache-a.pid
```

S'il n'existe pas, il faut le relancer, en tant que root :

```
service carbon-cache start
```

S'il fonctionne, vérifier qu'il écoute bien sur le port **2003** :

```
$ netstat -laputen | grep 2003
tcp 0 0 0.0.0.0:2003 0.0.0.0:* LISTEN 0 300518846 21989/python
```

Le numéro de processus (*ici **21989***) doit correspondre à celui du démon, dans le cas contraire, un autre processus a réservé le port et carbon-cache ne peut pas le prendre.

S'il n'est pas possible de se connecter au carbon-cache, vérifier que le port est ouvert dans le pare-feu.

Les logs de **carbon-cache** sont situés dans son espace de stockage **/opt/graphite/storage/log/carbon-cache/carbon-cache-a.**

Ils sont composés de 3 fichiers de logs :

- **console.log**: log principal du daemon
  - *16/06/2020 14:58:34 :: Log opened.* : démarrage du daemon
  - *16/06/2020 14:58:30 :: Sorted 378 cache queues in 0.000253 seconds* : fonctionnement normal du démon qui toutes les secondes vérifie son cache de données
  - *16/06/2020 14:58:33 :: Server Shut Down.* : arrêt du daemon
- **query.log**: log listant les connexions entrantes
  - *16/06/2020 14:13:24 :: 49.235.118.98:46670 connected* : connexion d'un démon se connectant au cache de données, typiquement grafana
  - *16/06/2020 14:13:24 :: 49.235.118.98:46670 disconnected* : déconnexion du cache de données
- **listener.log**: log listant les connexions entrantes :
  - *16/06/2020 08:09:16 :: MetricPickleReceiver connection with 185.209.0.165:2791 established* : connexion d'un nouvel écrivain
  - *16/06/2020 08:09:16 :: MetricPickleReceiver connection with 185.209.0.165:2791 closed cleanly* : déconnexion d'un écrivain

Vérification du pare-feu

S'il n'est toujours pas possible de se connecter au carbon-cache, vérifier que le port est ouvert dans le firewall.

Dès que le Graphite ne tourne pas sur la même machine

Exemple avec firewalld

Si **firewalld** est activé sur la machine qui fait tourner le serveur de métrologie et si le serveur de métrologie ne tourne pas sur la machine du Broker (*c'est-à-dire que la valeur du paramètre **host** est différente de localhost, 127.0.0.1, de l'IP ou nom de la machine qui fait tourner le Broker*), il faut autoriser les connexions vers le serveur de métrologie.

Sur la machine qui fait tourner le serveur de métrologie, vérifier que le port souhaité est ouvert (*par exemple 2003 pour carbon-cache*) dans firewalld :

```
firewall-cmd --list-ports
```

Exemple de retour et interprétation :

Exemple de résultat
`80/tcp 7763/tcp 7765/tcp 7766/tcp 7767/tcp 7768/tcp 7769/tcp 7770/tcp 7771/tcp 7772/tcp 7773/tcp 7777/tcp 7780/tcp 50000/tcp`

Dans cet exemple, le port 2003/tcp n'est pas listé, il est donc bloqué.

Pour ajouter un port dans firewalld :

```
firewall-cmd --add-port=2003/tcp
firewall-cmd --runtime-to-permanent
```

Vérification d'Apache, démon répondant aux requêtes de lectures

RHEL / CentOS 7 ou RHEL / Alma / Rocky 8

C'est le démon **Apache** qui héberge l'application répondant aux requêtes de lecture. Il faut des processus **httpd** ainsi que **wsgi:graphite** pour avoir le bon fonctionnement d'apache :

```
ps -fu apache |egrep 'httpd|wsgi'
apache 2194 31002 0 15:07 ? 00:00:00 /usr/sbin/httpd -DFOREGROUND
apache 6144 31002 1 15:09 ? 00:00:00 (wsgi:graphite) -DFOREGROUND
apache 31003 31002 0 15:06 ? 00:00:00 (wsgi:graphite) -DFOREGROUND
apache 31004 31002 0 15:06 ? 00:00:00 (wsgi:graphite) -DFOREGROUND
apache 31005 31002 0 15:06 ? 00:00:00 (wsgi:graphite) -DFOREGROUND
apache 31007 31002 0 15:06 ? 00:00:00 (wsgi:graphite) -DFOREGROUND
apache 31008 31002 0 15:06 ? 00:00:00 /usr/sbin/httpd -DFOREGROUND
apache 31009 31002 0 15:06 ? 00:00:00 /usr/sbin/httpd -DFOREGROUND
apache 31011 31002 0 15:06 ? 00:00:00 /usr/sbin/httpd -DFOREGROUND
apache 31012 31002 0 15:06 ? 00:00:00 /usr/sbin/httpd -DFOREGROUND
apache 31013 31002 0 15:06 ? 00:00:00 /usr/sbin/httpd -DFOREGROUND
```

Si ce n'est pas démarré, il faut lancer :

```
systemctl start httpd
```

Les logs d'apache pour graphite sont dans les répertoires **/var/log/graphite** et **/opt/graphite/storage/log/webapp** :

- **exception.log** : doit être vide, dans le cas contraire une erreur majeure est survenue
- **info.log** : log principal d'activité de la partie application de graphite, avec notamment les mises à jour du mapping entre nomuuids nécessaire par grafana
- **graphite-webapp.error.log**: toutes les erreurs d'accès aux pages, équivalent des erreurs 404 ou 500 dans apache
- **graphite-webapp.access.log**: log des accès réussis aux pages, équivalent des logs 200 d'apache

Ces fichiers sont définis dans le fichier **/etc/httpd/conf.d/graphite.conf** (*Attention, il ne faut pas modifier le graphite.conf car il est écrasé à chaque mise à jours* **).**

Debian 13

C'est le démon **Apache** qui héberge l'application répondant aux requêtes de lecture. Il faut des processus **apache2** ainsi que **wsgi:graphite** pour avoir le bon fonctionnement d'apache :

```
ps -fu apache |egrep 'apache2|wsgi'
www-data  673308     745  0 00:23 ?        00:00:00 /usr/sbin/apache2 -k start
www-data  673309     745  0 00:23 ?        00:00:00 /usr/sbin/apache2 -k start
www-data 1338632     745  0 14:10 ?        00:00:00 (wsgi:graphite)   -k start
www-data 1342656     745  0 14:15 ?        00:00:00 (wsgi:graphite)   -k start
www-data 1342657     745  0 14:15 ?        00:00:00 (wsgi:graphite)   -k start
www-data 1342658     745  0 14:15 ?        00:00:00 (wsgi:graphite)   -k start
www-data 1342659     745  0 14:15 ?        00:00:00 (wsgi:graphite)   -k start
```

Si ce n'est pas démarré, il faut lancer :

```
systemctl start apache2
```

Les logs d'apache pour graphite sont dans les répertoires **/var/log/graphite** et **/opt/graphite/storage/log/webapp** :

- **exception.log** : doit être vide, dans le cas contraire une erreur majeure est survenue
- **info.log** : log principal d'activité de la partie application de graphite, avec notamment les mises à jour du mapping entre nomuuids nécessaire par grafana
- **graphite-webapp.error.log**: toutes les erreurs d'accès aux pages, équivalent des erreurs 404 ou 500 dans apache
- **graphite-webapp.access.log**: log des accès réussis aux pages, équivalent des logs 200 d'apache

Ces fichiers sont définis dans le fichier **/etc/apache2/sites-available/graphite.conf** (*Attention, il ne faut pas modifier le graphite.conf car il est écrasé à chaque mise à jours*)**.**