Base de métrologie (Graphite)

Sommaire

Les métriques dans Shinken

Shinken Entreprise effectue un certain nombre de vérifications sur des hôtes et clusters, appelées "checks". Chaque check peut lors de son exécution extraire une donnée de performance qui est ensuite traitée par Shinken. Ces données peuvent être de tous types:

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

Les métriques sont présentées par des nombres flottants, qui pourront être ensuite consultés sous forme de graphes dans une interface.

Stockage des métriques

Dans Shinken Entreprise, les métriques sont stockées dans une base de données Graphite (https://graphiteapp.org/).

Les données de métrologie sont enregistrées dans Graphite par l'intermédiaire du module "Graphite-Perfdata", placé sur le Broker. Le Broker envoie les données au démon carbon (partie de Graphite), qui gère le stockage de ces données.

Les métriques sont physiquement rangés dans le répertoire "/opt/graphite/storage/whisper"

Consultation des métriques

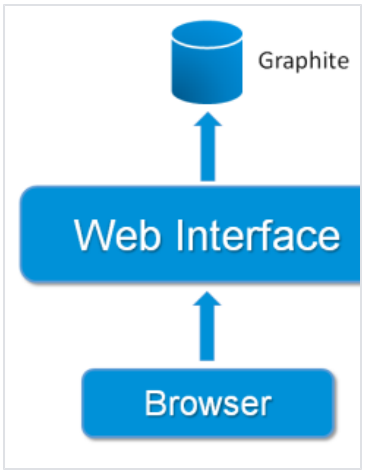
Interface de visualisation

L'accès aux métriques via l'interface de Visualisation est par défaut disponible et ne demande pas de configuration de la part de l'utilisateur.

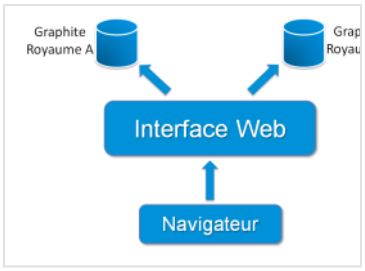
Les métriques sont accessibles de 2 manières différentes:

- Dans un tableau de bord, avec le Widget Graphique
- Directement sur un hôte/cluster ou un check dans l'Onglet Graphiques

Pour accéder aux métriques stockées dans Graphite, Shinken utilise Apache. Lorsqu'un utilisateur dans l'interface de Visualisation demande la visualisation d'une métrique, l'interface Web 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.



Outils externes

Ouvrir l'accès à Graphite aux outils externes

Par défaut, par mesure de sécurité Graphite est accessible seulement localement. Un serveur externe qui envoie une requête à Graphite se verra refuser l'accès.

Pour autoriser des serveurs externes à accéder à Graphite, il faut modifier la configuration d'Apache qui est responsable de la mise à disposition de Graphite au monde extérieur :

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

```
<VirtualHost 127.0.0.1:80>


à remplacer par


Listen PORT
<VirtualHost IP_INTERFACE:PORT>
```

avec:

- **IP_INTERFACE** : à remplacer par 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*). Utilisez * pour écouter sur toutes les interfaces
- **PORT :** Port d'écoute à utiliser. La directive "Listen" n'est pas obligatoire si le port par défaut 80 est utilisé.

- Plus d'informations sont disponibles sur les possibilités de configuration d'Apache sur la page de documentation suivante: https://httpd.apache.org/docs/2.4/en/bind.html
- Plus d'informations sur le changement du port dans ce chapitre : Changer le port de Graphite

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

```
service httpd restart
```

Changer le port de Graphite

Pour changer le port de graphite, il faut modifier deux fichiers :

- Le fichier de configuration de Graphite : **/etc/httpd/conf.d/graphite.conf**
- Le fichier de configuration d'Apache : **/etc/httpd/conf/httpd.conf**

Changer le port dans la configuration de Graphite

Par défaut, Graphite écoute sur le port 80, pour changer ce port, il faut aller voir dans le fichier **/etc/httpd/conf.d/graphite.conf**, là où se trouve la partie "VirtualHost".

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

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

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

Pour ouvrir un autre port, il suffit changer *80* par le port souhaité (*par exemple 8080*).

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

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

Plus d'informations sur l'adresse IP du VirtualHost dans ce chapitre : Ouvrir l'accès à Graphite aux outils externes

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

```
service httpd restart
```

Avant de redémarrer Apache, nous vous conseillons de changer aussi le port d'Apache afin de ne pas le redémarrer deux fois (*chapitre Changer le port dans la configuration d'Apache*).

Changer le port dans la configuration d'Apache

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, trouvez la partie où est écrit *Listen 80* :

/etc/httpd/conf/httpd.conf

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

Pour changer le port, il suffit donc de modifier le port d'écoute. 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 votre VirtualHost sera accessible depuis l'extérieur du server (*voir chapitre Changer le port dans la configuration de Graphite*).

/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*).

```
service httpd restart
```

Avant de redémarrer Apache, nous vous conseillons de changer aussi le port de Graphite afin de ne pas redémarrer Apache deux fois (*chapitre Changer le port dans la configuration de Graphite*).

Erreur lors du démarrage d'Apache

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

Il est possible que ce soit SELinux (*ou votre pare-feu si vous en avez un*) qui bloque le port que vous ayez choisi.

Dans le cas où ça serait SELinux, deux choix s'offrent à vous :

- changer de port;
- dire à SELinux d'accepter le port que vous avez choisi.

Pour avoir la liste complète des ports acceptés par les règles de SELinux pour vous pouvez lancer cette commande :

```
semanage port -l
```

Si vous voulez filtrer ces résultats pour les règles http, vous pouvez utiliser grep :

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

Si vous voulez changer le port que vous avez mis pour Graphite, vous pouvez choisir parmi ceux listés par la commande précédente.

Avant de continuer et ajouter une exception dans une règle de SELinux, il faut prendre en compte deux choses :

- cela va demander de redémarrer tout le système;
- cela peut comporter des risques en termes de sécurité. Vous pouvez toujours changer le port que vous avez choisi par ceux listés par les commandes précédentes.

Si vous souhaitez ajouter un port pour la règle *http_port_t* vous pouvez lancer cette commande :

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

Pour que les changements soient pris en compte, il faut redémarrer **le serveur**.

```
reboot
```

Pour plus d'informations, référez-vous à 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

Correspondance ID 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 NOMID.
- Pour résoudre ce problème, Shinken a mis une passerelle pour ne pas perturber les outils externes.
    - Par défaut les appels à Graphite renvoient les noms comme clef des métriques au monde extérieur.
    - Le broker et ses modules interrogent graphite avec un paramètre additionnel qui permet d'accéder aux métriques via les UIID.

Graphite a besoin de mettre à jour sa table de correspondance des noms pour les nouveaux checks et hôtes et dans le cas des renommages.

- Cette correspondance est contenue dans la base de données Mongo , dont l'accès est configuré dans Graphite dans le fichier **/opt/graphite/conf/mongodb.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

Configuration de l'accès à MongoDB

Pour se connecter au serveur Mongo, 2 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 sur toujours possible (voir Sécurisation des connexions aux bases MongoDB) mais bien plus complexe à mettre en place. La méthode de connexion par SSH est donc 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 Mongo (*/etc/mongod.conf*), assurez-vous que le paramètre "*bind_ip*" est positionné pour n'écouter que sur l'interface locale:
          `bind_ip= 127.0 . 0.1`

- Pour que graphite cré le tunnel, il faut activer les options suivantes (dans **/opt/graphite/conf/mongodb.conf**):

Nom de clé	Valeur par défaut	Description
URI	mongodb://**ADRESSE-SERVEUR-MONGO**/?w=1&fsync=false	URI du serveur Mongo L'adresse de la base Mongo à utiliser est l'adresse de la machine qui contient la base la plus complète des SLAs (généralement le broker central)
DATABASE	shinken	Nom de la base SLA sur le serveur Mongo
USE_SSH_TUNNEL	0	Activer la connexion à Mongo par Tunnel SSH

SSH_USER	shinken	User utilisé pour se connecter au serveur Mongo
SSH_KEY FILE	/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 vous 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	5	Timeout utilisé pour tester le tunnel SSH avant de lancer la connexion mongo

.

- Le tunnel SSH va permettre au module de se connecter comme si ses requêtes étaient local au serveur mongo (en 127.0.0.1).
    - Depuis le serveur hébergeant Graphite, assurez-vous que les clés publiques SSH de l'utilisateur lançant le démon Apache (par défaut "apache") sont autorisées sur le serveur hébergeant Mongo :
        - Connectez-vous avec le user lançant le démon sur le serveur Shinken
        - Générez la paire de clés SSH si nécessaire
        - Copiez la clé publique sur le serveur mongo

```
root@serveur_shinken # su - apache
shinken@serveur_shinken $ ssh-keygen
shinken@serveur_shinken $ ssh-copy-id user_distant@serveur_mongo
[...]
shinken@serveur_shinken $ ssh user_distant@serveur_mongo
user_distant@serveur_mongo $
```

    - Vous pouvez également utiliser la clé ssh du user "shinken" ; dans ce cas il vous faut copier la clé privé du user shinken (/var/lib/shinken/.ssh/id_rsa) à un emplacement accessible par le démon Apache, avec les droits d'accès permettant au user "apache" de lire ce fichier.
- Cette manipulation est aussi nécessaire dans le cas où la base mongoDB est sur le même serveur que le module SLA, même si le tunnel est ouvert localement.

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 manuelles 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 du bon fonctionnement de graphite

Vérification de carbon-cache, le demon écrivain

Pour vérifier que le daemon **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 bien évidement le relancer, en tant que root:

```
/etc/init.d/carbon-cache start
```

S'il fonctionne, vérifiez 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 daemon, dans le cas contraire, un autre processus a réservé le port et carbon-cache ne peux pas le prendre.

Les logs de carbon-cache sont situés dans son espace de stockage **/opt/graphite/storage/log/carbon-cache/carbon-cache-a.** Il est composé 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 daemon qui toute 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 daemon 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 nouveau écrivain
  - *16/06/2020 08:09:16 :: MetricPickleReceiver connection with 185.209.0.165:2791 closed cleanly* : déconnexion d'un écrivain

Vérification d'Apache, demon répondant aux requêtes de lectures

C'est le demon **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 lancé, il faut lancer:

```
service httpd start
```

Les logs d'apache pour graphite sont définis dans le fichier **/etc/httpd/conf.d/graphite.conf** (Attention, il ne faut pas modifier ce fichier qui est écrasé lors des mises à jours) et sont dans le répertoire **/var/log/graphite**:

- **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 notament 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

Gestion de l'espace disque de graphite

Espaces des logs

Les logs contenus dans **/var/log/graphite** ainsi que **/opt/graphite/storage/log/carbon-cache/carbon-cache-a** ont déjà une rotation et ne vont pas grossir indéfiniment.

Concernant les logs de **/opt/graphite/storage/log/carbon-cache/carbon-cache-a** les logs de plus de **7** jours sont supprimés via le script **/etc/cron.daily/graphite-clean** qui, comme son chemin le spécifie, tourne une fois par jour.

Espaces des métriques

Les métriques contenu dans **/opt/graphite/storage/whisper** ne sont pas automatiquement supprimés lors de la suppression des hôtes et/ou des checks. Il est possible cependant d'automatiquement supprimer les métriques qui n'ont pas été mis à jour depuis très longtemps afin de récupérer l'espace disque, à définir dans **/etc/crontab**, ici avec une suppression des métriques non mis à jour depuis **90** jours, et lancés à **3h01** chaque jour:

```
1 3 * * * root find /opt/graphite/storage/whisper -name "*.wsp*" -type f -mtime +90 | xargs /bin/rm -f
```