

Haute disponibilité de la base de métrologie (Graphite)

Sommaire

- [Introduction](#)
- [Architecture mise en place](#)
- [Procédure de configuration](#)
 - [Installation de Shinken](#)
 - [Mise en place du démon carbon-relay](#)
 - [Autorisation des connexions à Graphite](#)
 - [Modification de la configuration Shinken pour l'utilisation du cluster Graphite](#)
 - [Redémarrage de Graphite et Shinken](#)
- [Vérification du firewall \(est ce que vous avez accès \)](#)
 - [Si vous avez firewalld \(firewall par défaut de la Redhat / AlmaLinux \)](#)
 - [Vérifier que les ports sont ouvert](#)
 - [Sur les serveurs carbon-relay](#)
 - [Sur les serveurs carbon-cache](#)
- [Passer le cluster Graphite en HTTPS](#)
- [Comportement de Shinken avec un cluster Graphite](#)
- [Outils externes \(Grafana \) et relation UUID nom](#)

Introduction

La base de métrologie de Shinken peut être fortement sollicitée sur des architectures supervisant un grand nombre d'éléments. Sa disponibilité est également importante puisqu'elle peut être source de données pour des outils externes à Shinken (*Grafana par exemple*) pouvant être mis à disposition.

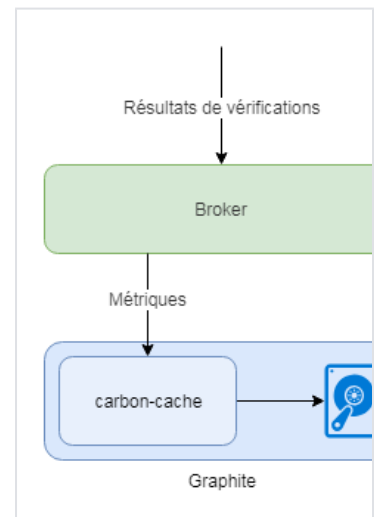
L'objectif de cette page de documentation est d'expliquer de manière détaillée la procédure nécessaire pour la mise en place d'un cluster Graphite. Cela permet d'augmenter la disponibilité des données et d'éviter la perte de données en cas d'incident sur la machine stockant les métriques.

Architecture mise en place

Dans une installation Shinken classique, les métriques sont stockées dans une base de données Graphite par le Broker.

Chaque résultat de vérification contenant des données de métrologie est analysé par le Broker. Ces métriques sont enregistrées dans Graphite par le Broker grâce au module Graphite-Perfdata.

Dans Graphite, le nom du démon responsable de l'enregistrement des métriques sur le disque est "carbon-cache".



Pour transformer cette architecture dans le but de la rendre hautement disponible, on voudra faire en sorte que les données soient répliquées lors de leur écriture. On utilise alors plusieurs démons "carbon-cache" qui enregistrent chacun les métriques qu'on leur envoie sur des machines différentes, et un démon "carbon-relay" qui s'occupe de distribuer l'enregistrement des métriques aux démons "carbon-cache".

L'architecture qu'on obtient à la fin de la procédure décrite dans cette documentation correspond à celle présente sur le schéma ci-contre.

Notre cluster Graphite est constitué d'au moins 2 machines distinctes (*idéalement 3 pour répartir les performances*). Chaque démon "carbon-cache" est sur une machine distincte, qui permettra de stockage des métriques. Le démon "carbon-relay" reçoit l'ensemble des métriques envoyées par le Broker. Il se charge ensuite d'envoyer les ordres d'écritures des métriques aux démons "carbon-cache".



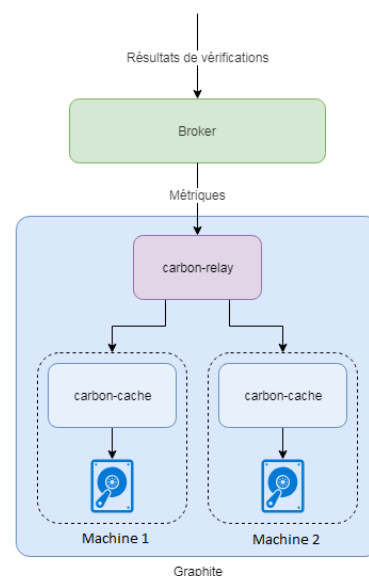
Le système mis en place ici est de la réplication, et non de la répartition.

Les démons "carbon-cache" stockent chacun l'ensemble des métriques envoyées par le Broker. Il faut donc que chaque machine qui héberge un démon "carbon-cache" soit capable d'enregistrer l'intégralité des métriques gérées par le Broker.



Pour une version de Shinken antérieure à la 02.08.01.09, il existe des incompatibilités entre un Graphite installé sur une CentOS 7 ou RedHat 7 et un Graphite installé sur une RedHat 8 ou Alma8. En raison de cette incompatibilité, il n'est pas possible de monter un cluster Graphite avec des nœuds sur des versions de distributions 7 et 8. Il est par contre possible d'avoir des nœuds dans chacune des distributions correspondant à la même version 7 ou 8.

Depuis la version 02.08.01.09, le Graphite installé est désormais compatible. Les nœuds d'un même cluster peuvent donc être installés sur n'importe quelle distribution supportée par Shinken.



Procédure de configuration

Avant de commencer, voici un résumé des différentes étapes nécessaires pour la configuration du cluster Graphite:

- Installation de Shinken.
- Mise en place du démon "carbon-relay".
- Autorisation des connexions à Graphite.
- Modification de la configuration Shinken pour l'utilisation du cluster Graphite.
- Redémarrage de Graphite et Shinken.

Installation de Shinken

La première étape dans la mise en place d'un cluster Graphite est bien entendu l'installation de Graphite. Comme les autres dépendances de Shinken, il faut que la version de Graphite installée soit celle fournie avec l'installateur Shinken. En effet, certaines modifications ont été faites sur Graphite pour l'intégration avec Shinken, ce qui rend incompatibles les versions de Graphites qui auraient pu être installées auparavant via d'autres installateurs.

Sur chaque machine qui compose le cluster Graphite, il faudra alors procéder à une installation de Shinken.

La procédure d'installation de Shinken est décrite une page de documentation dédiée (voir la page [Guide d'installation et de mise à jour](#)).

Mise en place du démon carbon-relay

Une fois Shinken et Graphite installés, il faut procéder d'abord à la configuration de Graphite.

On commence par dire à Graphite qu'on utilise la fonctionnalité de relay via le fichier `/opt/graphite/conf/carbon.conf` (copier le fichier d'exemple `carbon.conf.example` si besoin).

Dans la section "[relay]", on modifie la variable "DESTINATIONS" dans laquelle on décrit la liste des démons **carbon-relay** qui constituent le cluster Graphite.

`/opt/graphite/conf/carbon.conf`

```
[relay]
...
<autres options>
...

DESTINATIONS = adresse_carbon_cache_1:2004, adresse_carbon_cache_2:2004
```

On configure ensuite le relay pour lui spécifier les adresses des carbon-cache pour l'écriture et les adresses à utiliser pour la lecture des métriques.

Cette configuration se fait via le fichier `/opt/graphite/conf/relay-rules.conf` (copier le fichier d'exemple `relay-rules.conf.example` si besoin) :

```
[default]
default = true
destinations = adresse_carbon_cache_1:2004,adresse_carbon_cache_2:2004
read_destinations = adresse_carbon_cache_1:80,adresse_carbon_cache_2:80
```

Par défaut, l'installateur Shinken met en place le fichier d'initialisation pour le démon carbon-relay mais ne lui attribue pas les droits d'exécution. Pour pouvoir le démarrer en tant que service du système comme les autres démons, il va falloir lui donner le droit de s'exécuter. Pour cela, lancer la commande suivant sur la machine qui va héberger le démon carbon-relay :

Donner les droits d'exécution

```
chmod +x /etc/init.d/carbon-relay
```

À la fin de cette étape de configuration, le relais est correctement configuré en ce qui concerne l'écriture des données. Pour la lecture des métriques, il faut encore autoriser la lecture des métriques, ce qui est l'objectif de l'étape suivante.

Autorisation des connexions à Graphite

Par défaut, Graphite autorise la lecture des métriques seulement depuis une interface locale, pour des raisons de sécurité. Il faut alors sur chaque machine qui héberge un démon carbon-relay, modifier les réglages d'Apache pour autoriser la lecture sur des machines externes.

Sur chaque carbon-relay et carbon-cache, on trouve dans le fichier `/etc/httpd/conf.d/graphite.conf` la ligne suivante:

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

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

Cette ligne permet de dire à Apache qu'il faut écouter les requêtes de lecture des métriques sur l'interface locale (`127.0.0.1`) et le port 80 seulement. Pour écouter ces requêtes sur toutes les interfaces réseau de la machine, on remplacera cette ligne par la suivante:

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

```
<VirtualHost *:80>
```

On redémarre ensuite Apache pour prendre en compte les modifications :

```
systemctl restart httpd
```

Modification de la configuration Shinken pour l'utilisation du cluster Graphite

Comme vu dans la section présentant l'architecture haute disponibilité qu'on est en train de mettre en place, le Broker, et en particulier son module Graphite-Perfdata, doit maintenant envoyer ses informations au démon carbon-relay au lieu de les envoyer directement à un démon carbon-cache.

Il faut donc modifier la configuration actuelle de Shinken pour envoyer les métriques sur le démon carbon-relay plutôt qu'au démon carbon-cache.

Pour ça, on modifie la configuration du module Graphite-Perfdata dans le fichier `/etc/shinken/modules/graphite.cfg` :

```

# Metrology server parameters #

# Graphite writer ( carbon ) address
# IP address or FQDN of carbon-cache or carbon-relay instance to send metrics to
#
#         Default : localhost
#
broker__module_graphite_perfdata__writer__host      adresse_carbon_relay

# Graphite writer ( carbon ) port
# tcp port of carbon-cache or carbon-relay instance to send metrics to
#
#         Default : 2003
#
broker__module_graphite_perfdata__writer__port      2013

```

On change ici le port 2003 (*port par défaut du carbon-cache*) vers le port 2013 (*port par défaut du carbon-relay*).

Il faut également modifier les réglages de Shinken pour lui spécifier d'envoyer les requêtes de lecture des métriques sur le carbon-relay au lieu de les envoyer directement sur un démon carbon-cache.

Cette modification se fait directement dans les réglages du module WebUI, responsable de l'interface de Visualisation. Dans **/etc/shinken/modules/webui.cfg**, on modifie la section suivante :

/etc/shinken/modules/webui.cfg

```

[...]

graphite_backends      *=http://adresse_carbon_relay:80

[...]

```

Redémarrage de Graphite et Shinken

À ce stade, la configuration de Graphite a été modifiée pour utiliser un démon carbon-relay.

On a aussi effectué des modifications de configuration au niveau de Shinken pour prendre en compte ce changement d'architecture au niveau de Graphite.

La dernière étape est de redémarrer les différents démons pour prendre en compte ces changements de configuration.

- On commence par démarrer le démon carbon-relay Graphite:

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

- Ajouter carbon-relay au démarrage du serveur

```
systemctl enable carbon-relay
```

- On redémarre ensuite les démons carbon-cache sur leurs machines respectives

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

- On redémarre ensuite l'Arbiter pour prendre en compte les modifications de configuration de Shinken:

```
/etc/init.d/shinken-arbiter restart
```

Vérification du firewall (est ce que vous avez accès)

Si vous n'arrivez pas à vous connecter au carbon-cache / carbon-relay vérifier que le port est ouvert dans votre firewall.

Si vous avez firewalld (firewall par défaut de la Redhat / AlmaLinux)

Vérifier que les ports sont ouvert

Voici comment vérifier que les ports des carbon-relay / carbon-cache sont ouvert sur leurs machines :

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

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, aucun de port de carbon-relay (2013/tcp) et carbon-cache (2004/tcp) sont présents, ils sont donc bloqués.

Sur les serveurs carbon-relay

Si le port du **carbon-relay** n'est pas ouvert, il faut alors autoriser le trafic vers ce port.

Cela peut être fait avec les commandes suivantes :

```
firewall-cmd --add-port=2013/tcp
firewall-cmd --runtime-to-permanent
```

Sur les serveurs carbon-cache

Si le port du **carbon-cache** n'est pas ouvert, il faut alors autoriser le trafic vers ce port.

Cela peut être fait avec les commandes suivantes :

```
firewall-cmd --add-port=2004/tcp
firewall-cmd --runtime-to-permanent
```

Passer le cluster Graphite en HTTPS

Les étapes suivantes permettent d'activer le HTTPS d'un cluster Graphite :

- Sur chaque nœud du cluster (carbon-relay et carbon-cache), activez le HTTPS de Graphite (voir la page [Base de métrologie \(Graphite \) section Activation de HTTPS pour Graphite](#)).
- Activer HTTPS pour les communications intra-cluster. Sur la machine du **carbon-relay** dans le fichier de configuration `/opt/graphite/webapp/graphite/local_settings.py` (copier le fichier d'exemple `local_settings.py.example` si besoin), dé-commenter la clé `INTRACLUSTER_HTTPS` et la passer à True

`/opt/graphite/webapp/graphite/local_settings.py`

```
[ ... ]

INTRACLUSTER_HTTPS = True

[ ... ]
```

- Ne pas oublier de changer pour le **carbon-relay** les ports des carbon-cache à utiliser pour la lecture des métriques (443 désormais au lieu de 80).

Cette configuration se fait via le fichier **/opt/graphite/conf/relay-rules.conf** (*copier le fichier d'exemple **relay-rules.conf.example** si besoin*) :

```
[default]
default = true
destinations = adresse_carbon_cache_1:2004,adresse_carbon_cache_2:2004
read_destinations = adresse_carbon_cache_1:443,adresse_carbon_cache_2:443
```

- Si ce n'est pas déjà fait, préciser dans le fichier de configuration de la WebUi (**/etc/shinken/modules/webui.cfg**) que le cluster Graphite communique en HTTPS. Il faut ensuite redémarrer l'Arbiter pour prendre en compte la modification.

/etc/shinken/modules/webui.cfg

```
[ ... ]

graphite_backends                               *=https://CARBON_RELAY_SERVER:443

[ ... ]
```

- Assurez-vous que les certificats sont acceptés par chaque nœud. Le serveur où se trouve le carbon-relay doit accepter le certificat Graphite des serveurs où se trouvent les carbon-cache. Les serveurs où se trouvent les carbon-cache doivent accepter le certificat Graphite du serveur carbon-relay. Si vos certificats ne sont pas reconnus, vous pouvez les ajouter à la liste des certificats de confiance de votre système. Pour ce faire, vous pouvez ajouter le certificat de Graphite ou celui de l'autorité de certification dans **/etc/ssl/certs/ca-bundle.trust.crt**. Les deux commandes suivantes permettent de rajouter un certificat dans les certificats de confiance du système.

```
cp your_certificate.crt /etc/pki/ca-trust/source/anchors/
update-ca-trust
```

- Sur l'ensemble des machines redémarrer le démon httpd

```
service httpd restart
```

- Redémarrer le démon carbon-relay Graphite:

```
/etc/init.d/carbon-relay restart
```

Comportement de Shinken avec un cluster Graphite

Comme vu dans la procédure de configuration, la modification de configuration côté Shinken pour utiliser le cluster Graphite est minimale : on se contente de changer les adresses utilisées pour la lecture et l'écriture des métriques pour pointer sur le démon carbon-relay au lieu d'un démon carbon-cache.

La haute disponibilité de Graphite est donc gérée entièrement par Graphite, qui s'occupe de la réplication des données entre les démons carbon-cache et de la lecture des données. Cette modification d'architecture est invisible pour les utilisateurs.

Outils externes (*Grafana*) et relation UUID nom

En cas d'utilisation d'outils externes (*comme Grafana par exemple*) 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 page [Base de métrologie \(Graphite \)](#) section *Correspondance ID Nom de l'élément*)