

Haute disponibilité de la base de métrologie (Graphite)

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.

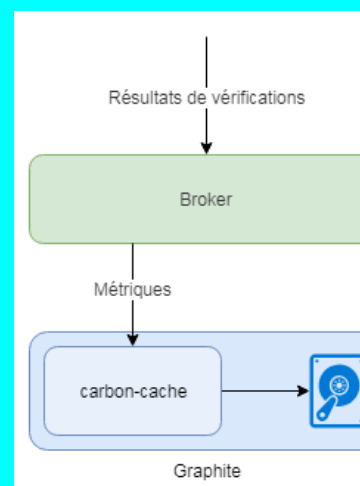
Introduction
Architecture mise en place
Procédure de configuration
Installation de Shinken
Mise en place du démon carbon-relay
Autorisation des connexions à Graphite
Modification de la configuration Shinken pour l'utilisation du cluster Graphite
Redémarrage de Graphite et Shinken
Comportement de Shinken avec un cluster Graphite

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 sont analysées 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 pour la rendre hautement disponible, on veut 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".

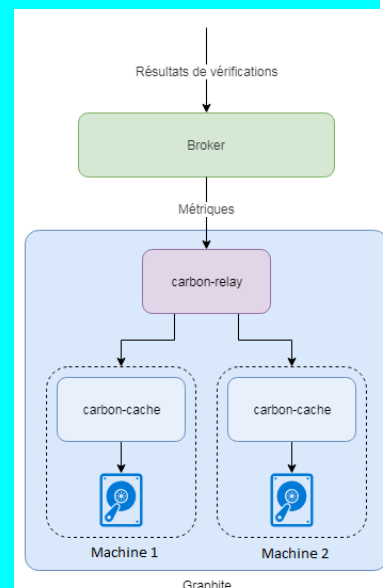
L'architecture qu'on obtient à la fin de la procédure décrite dans cette documentation correspond à celle présente sur le schéma ci-contre.

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 envoyés 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.



Procédure de configuration

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

- 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 incompatible 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 dans la page de documentation dédiée: [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 l'écriture 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 ne met pas en place de fichier d'init pour le démon carbon-relay. Pour pouvoir le démarrer en tant que service du système comme les autres démons, il va falloir mettre en place ce fichier.

Pour cela, copier le fichier suivant vers `/etc/init.d/carbon-relay` sur la machine qui va héberger le démon carbon-relay: [carbon-relay](#)

A 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 n'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, 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:

```
# Sur CentOS 6
service httpd restart
# Sur CentOS 7
systemctl restart httpd
```

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

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

Il faut donc modifier la configuration de Shinken pour envoyer les métriques sur le démon carbon-relay.

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

```
##### Graphite address #####
# host:  graphite server address (ip or fqdn)
host      adresse_carbon_relay

# port:  tcp port of the graphite server
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 le 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

[...]

```
##### Metrology access #####  
# Multi-realm graphite parameter  
graphite_backends      *:adresse_carbon_relay
```

[...]

Redémarrage de Graphite et Shinken

A 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
```

- On redémarre ensuite les démons carbon-cache sur leur 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
```

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.