Haute disponibilité de la base de métrologie (Graphite) avec keepalived

Sommaire

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 des utilisateurs.

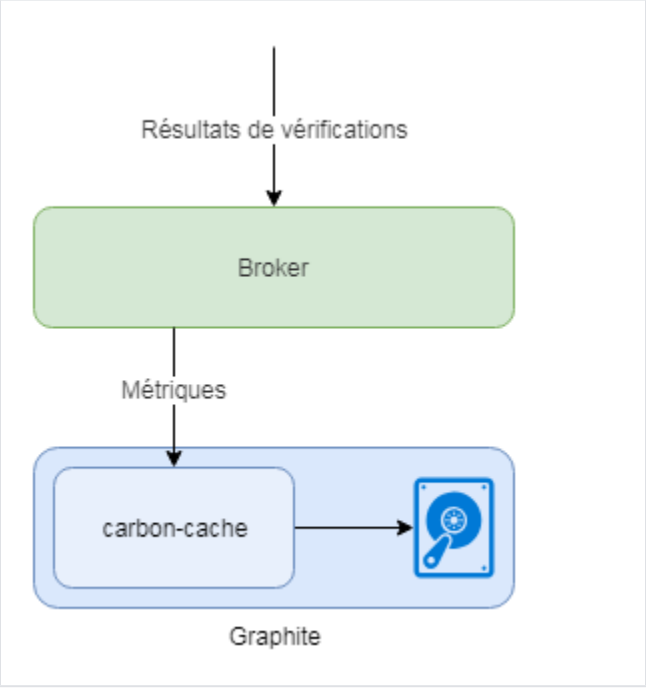
L'objectif de cette page de documentation est de détailler la procédure pour la mise en place d'un cluster Graphite. Ce qui permettra d'augmenter la disponibilité des données et d'éviter la perte de données en cas d'incident sur la machine stockant les métriques.

Architecture mise en place

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

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

Du côté de 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, les données seront répliquées lors de leur écriture.

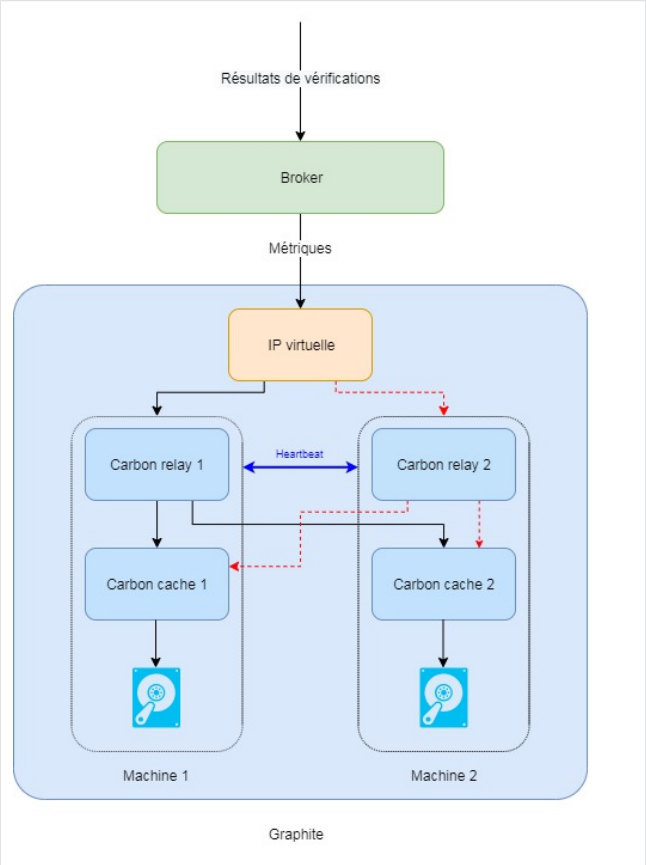
Le logiciel **Keepalived** permettra de créer une adresse IP virtuelle qui, en cas de problème sur le serveur principal, gèrera une bascule sur le serveur secondaire automatiquement.

Des démons "**carbon-relay**" sur le serveur principal ET le serveur secondaire, assureront un enregistrement des métriques sans interruption, même en cas de bascule.

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

Le cluster Graphite est constitué d'au moins 2 machines distinctes.

- Le démon "*carbon-cache*" est sur toutes machine du cluster, il assurera le stockage des métriques.
- Le démon "carbon-relay" est aussi sur toutes les machines du cluster, il fera suivre les métriques à tous les "carbon-cache" constituant le cluster.
- Sur la machine possédant l'adresse IP virtuelle, le démon "*carbon-relay*" reçoit effectivement 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.

⚠ **Pour que le cluster Graphite fonctionne correctement, il est nécessaire de l'installer sur des machines utilisant le même système d'exploitation.**

Procédure de configuration

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

- Installation de Shinken
- Installation et configuration de Keepalived
- 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'installeur Shinken. En effet, certaines modifications ont été faites sur Graphite pour l'intégration avec Shinken, ce qui rend incompatible les versions de Graphite qui auraient pu être installées auparavant via d'autres installeurs.

Sur chaque machine qui compose le cluster Graphite, il faudra 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).

Installation et configuration de Keepalived

Sur chaque machine constituant le cluster Graphite

Installer Keepalived :

RHEL / CentOS 7 ou RHEL / Alma / Rocky 8 ou RHEL / Alma / Rocky 9

- ```
  yum install keepalived
  ```

Debian 13

- ```
  apt install keepalived
  ```

Éditer le fichier de configuration de Keepalived

/etc/keepalived/keepalived.conf

```
! Configuration File for keepalived

vrrp_instance NOM_DU_SERVEUR {
# Priorité utilisée lors de l'élection d'un nouveau maître. C'est la priorité la plus élevée qui gagne
l'élection
  priority 150
# ID du routeur virtuel. Doit être unique sur l'ensemble des noeuds
  virtual_router_id 255
# Option qui a pour conséquence qu'un master qui renaît ne reprendra pas l'IP si son SLAVE l'a
  nopreempt
# Interface d'écoute et d'échange des paquets multicast VRRP
  interface enp0s3
# L'état de santé des cibles est vérifié toutes les 5 secondes
  advert_int 5
# Interface réseau commune aux membres
  virtual_ipaddress {
        XXX.XXX.XXX.XXX/XX brd XXX.XXX.XXX.XXX dev enp0s3 scope global
  }
}
```

ATTENTION à modifier les informations suivantes :

- vrrp_instance NOM_DU_SERVEUR : Le nom associé à l'adresse virtuelle
- priority 150 : 150 pour le serveur principal et une valeur inférieur pour les secondaires
- interface enp0s3 : Le nom de l'interface réseau principale du serveur (*commande "ip addr" ou "ifconfig"*)
- virtual_ipaddress XXX.XXX.XXX.XXX/XX :
  - En 1er l'adresse virtuelle qui va être créée.
  - En 2ème le masque de sous réseau.
  - En 3ème l'adresse de broadcast du sous-réseau (*se termine en général par 255*).
  - En 4ème le nom de l'interface réseau du serveur.

Exemple d'un configuration avec 1 serveur principal et 1 serveur secondaire

MASTER : /etc/keepalived/keepalived.conf

```
! Configuration File for keepalived

vrrp_instance graphite-relay {
  priority 150
  virtual_router_id 255
  nopreempt
  interface enp0s3
  advert_int 5
  virtual_ipaddress {
        192.168.1.100/24 brd 192.168.1.255 dev
enp0s3 scope global
  }
}
```

SLAVE : /etc/keepalived/keepalived.conf

```
! Configuration File for keepalived

vrrp_instance graphite-relay {
  priority 100
  virtual_router_id 255
  nopreempt
  interface enp0s3
  advert_int 5
  virtual_ipaddress {
        192.168.1.100/24 brd 192.168.1.255 dev
enp0s3 scope global
  }
}
```

Redémarrer Keepalived sur tous les serveurs du cluster en commençant par le maître :

```
systemctl restart keepalived
```

Ainsi la commande "ip addr" sur le serveur maître devrait renvoyer l'ip du serveur et l'ip virtuelle précédemment créée :

```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:20:4a:2f brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.68/24 brd 192.168.1.255 scope global noprefixroute dynamic enp0s3
       valid_lft 75921sec preferred_lft 75921sec
    inet 192.168.1.100/24 brd 192.168.1.255 scope global secondary enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe20:4a2f/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Mise en place du démon carbon-relay

Sur tous les serveurs constituant le cluster Graphite :

Une fois Shinken, Graphite et Keepalived installés, il faut procéder à la configuration de Graphite.

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

On configure le relais en précisant les adresses des serveurs hébergeant les démons carbon-cache (*pour l'écriture et la lecture des métriques*).

/opt/graphite/conf/relay-rules.conf

```
[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
```

On configure ensuite le démon carbon-relay pour préciser la liste des carbon-cache constituant le cluster.

Pour cela éditer 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-cache** qui constituent le cluster Graphite.

/opt/graphite/conf/carbon.conf

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


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

Par défaut, l'installeur Shinken met en place un fichier de démarrage pour le démon carbon-relay sans les droits d'exécution. Pour permettre un démarrage en tant que service du système, il faut ajouter le droit d'exécution.
Sur la machine qui va héberger le démon carbon-relay, lancer la commande suivante :

Donner les droits d'execution

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

À ce stade de la configuration, le relais est correctement configuré en ce qui concerne l'écriture des données.

Il faut maintenant permettre la lecture des métriques.

Autorisation des connexions à Graphite

Par défaut, Graphite autorise la lecture des métriques seulement depuis l'interface locale (*127.0.0.1*), pour des raisons de sécurité.

Pour le fonctionnement en cluster, il faut modifier les réglages d'Apache pour autoriser la lecture depuis le réseau.

RHEL / CentOS 7 ou RHEL / Alma / Rocky 8 ou RHEL / Alma / Rocky 9

Sur chaque serveur constituant le cluster Graphite, dans le fichier **/etc/httpd/conf.d/graphite.conf** on trouve la ligne suivante:

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

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

Cette directive indique 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 la remplacera par :

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

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

On redémarre ensuite Apache pour prendre en compte les modifications :

```
systemctl restart httpd
```

Debian 13

Sur chaque serveur constituant le cluster Graphite, dans le fichier **/etc/apache2/sites-available/graphite.conf** on trouve la ligne suivante:

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

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

Cette directive indique 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 la remplacera par :

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

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

On redémarre ensuite Apache pour prendre en compte les modifications :

```
systemctl restart apache2
```

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

Écriture des métriques

L'envoi des métriques à Graphite est géré par le module Graphite-Perfdata du Broker.

Pour que les métriques soient envoyées au relais sur l'IP virtuelle plutôt qu'au démon carbon-cache, il faut modifier la configuration du module Graphite-Perfdata (*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      addresse_ip_virtuelle

#  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

- l'adresse du serveur destinataire des métriques à écrire par l'IP virtuelle configurée avec keepalived,
- le port 2003 (*port par défaut du carbon-cache*) vers le port 2013 (*port par défaut du carbon-relay*).

Lecture des métriques

Il faut également préciser à Shinken de passer par l'adresse IP virtuelle pour la lecture des métriques.

Cette modification se fait dans les paramètres 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        *:addresse_ip_virtuelle


[...]
```

Redémarrage de Graphite et Shinken

Les configurations de Graphite et de Shinken ayant été modifiées pour utiliser le cluster, il faut maintenant redémarrer les différents démons pour appliquer ces changements.

- On commence par démarrer le démon carbon-relay sur chaque machine constituant le cluster Graphite :

  ```
  service carbon-relay start
  ```

- On fait en sorte que ce démon soit relancé au démarrage des serveurs (*sur chaque machine constituant le cluster Graphite*) :

  ```
  systemctl enable carbon-relay
  ```

- On redémarre le démon carbon-cache sur tous les serveurs composants le cluster:

  ```
  service carbon-cache restart
  ```

- Enfin, on redémarre l'Arbiter pour prendre en compte les modifications de configuration de Shinken:

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

Vérification du firewall (*est-ce que l'accès à carbon-cache / carbon-relay est disponible*)

En cas de problème de connexion aux démons carbon-cache / carbon-relay vérifier que le port est ouvert dans le firewall.

Liste des ports utilisés par Graphite

La liste des ports utilisés par Graphite est la suivante :

Port	Serveur hébergeant	Fonctionnalité
2003	carbon-cache	Réception des métriques à écrite, au format texte clair.
2004	carbon-cache	Réception des métriques à écrire, au format binaire (*les données sont sérialisées*).
2013	carbon-relay	Réception des métriques à relayer, au format texte clair.
2014	carbon-relay	Réception des métriques à relayer, au format binaire (*les données sont sérialisées*).
80	carbon-relay et carbon-cache	Lecture des métriques (*protocole http*).
443	carbon-relay et carbon-cache	Lecture des métriques (*protocole https*).

Si firewalld est présent (***firewall activé par défaut sur RHEL / Alma / Rocky 8 et RHEL / Alma / Rocky 9***)

Vérifier que les ports de Graphite sont ouverts

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

```
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 des ports de carbon-relay (*2013/tcp*) et de carbon-cache (*2004/tcp*) ne sont présents, ils sont donc bloqués.

Autoriser le trafic vers les ports de Graphite

Sur tous les serveurs constituant le cluster :

Si le port du démon **carbon-relay** n'est pas ouvert, pour autoriser le trafic vers ce port, utiliser les commandes suivantes :

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

Si le port du **carbon-cache** n'est pas ouvert, pour autoriser le trafic vers ce port, utiliser les commandes suivantes :

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

Vérifier les autorisations pour les communications de keepalived

Pour vérifier que les démons **keepalived** puissent communiquer, lancer la commande suivante

```
firewall-cmd --list-all
```

La ligne **rich rules** indiquera si le protocol "vrrp" est autorisé.

Voici deux exemples de sorties,

- sur le premier l'autorisation est en place

Autorisation keepalived présente

```
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3 enp0s8
  sources:
  services: cockpit dhcpv6-client ssh
  ports: 80/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 3000/tcp
  protocols:
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
        rule protocol value="vrrp" accept
```

- sur le deuxième, elle ne l'est pas :

Autorisation keepalived absente

```
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3 enp0s8
  sources:
  services: cockpit dhcpv6-client ssh
  ports: 80/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 3000/tcp
  protocols:
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Autoriser les démons keepalived à communiquer

Si le protocole n'est pas autorisé par le firewall, pour autoriser le trafic pour **keepalived**, utiliser les commandes suivantes :

```
firewall-cmd --add-rich-rule='rule protocol value="vrrp" accept' --permanent
firewall-cmd --reload
```

Comportement de Shinken avec un cluster Graphite

La haute disponibilité de Graphite est entièrement gérée par Keepalived, puis Graphite 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 de Shinken.

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 tous les serveurs constituant le cluster,
- autoriser les connexions au serveur d'inventaire depuis les serveurs constituant le cluster.

(*voir la page* Base de métrologie (Graphite) *section Correspondance ID Nom de l'élément*)