

---

# Documentation de la mise en place de graphite et sont utilisation pour le monitoring de machine netapp

---

## Introduction

Ce document simple rédigé en Markdown a pour but de documenter la mise en place de graphite et sont utilisation pour le monitoring de machine netapp.

## Installation de Graphite

### Le proxy

Dans une premier temps nous allons ajouter le proxy en créant/éditant le fichier environnement :

```
nano /etc/environnement
```

insérez le contenu suivant : `httpproxy="http://nomproxy:mdp@nomdedomaine:port"`  
`httpsproxy="http://nomproxy:mdp@nomdedomaine:port"`

puis dans le terminal faites ceci :

```
export http_proxy="http://nom_proxy:mdp@nom_de_domaine:port"  
export https_proxy="http://nom_proxy:mdp@nom_de_domaine:port"
```

### Installation des paquets

On commence par les paquets RPM :

```
sudo yum update  
sudo yum install -y python-devel mod_wsgi pycairo dejavu-sans-fonts gcc git pytz python-memcached nc
```

Nous installons maintenant les paquets python :

```
cd /tmp  
curl -O https://bootstrap.pypa.io/get-pip.py  
python get-pip.py  
pip install pyparsing  
pip install 'Twisted<12.0'  
pip install 'django-tagging==0.3.6'  
pip install 'django<1.5'
```

On installe maintenant les composants de graphite depuis les sources :

```
cd /tmp  
git clone https://github.com/graphite-project/graphite-web.git  
git clone https://github.com/graphite-project/carbon.git  
git clone https://github.com/graphite-project/whisper.git  
cd whisper; git checkout 0.9.13-pre1; python setup.py install  
cd ../carbon; git checkout 0.9.13-pre1; python setup.py install  
cd ../graphite-web; python check-dependencies.py; git checkout 0.9.13-pre1; python setup.py install
```

On copie les fichiers de configuration depuis les exemples :

```
cd /opt/graphite/conf
cp carbon.conf.example carbon.conf
cp graphite.wsgi.example graphite.wsgi
cp storage-schemas.conf.example storage-schemas.conf
cp storage-aggregation.conf.example storage-aggregation.conf

cd /opt/graphite/webapp/graphite/
cp local_settings.py.example local_settings.py
cp /opt/graphite/examples/example-graphite-vhost.conf /etc/httpd/conf.d/graphite-vhost.conf
```

## Configuration de Carbon

On ouvre le fichier de configuration de carbon :

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

On change la ligne `MAXCREATESPER_MINUTE` de 50 à 600 pour permettre la création de plus de fichier de métrique par minute :

```
MAX_CREATES_PER_MINUTE = 600
```

On règle le cache de carbon (carbon-cache) pour qu'il démarre automatiquement et on le démarre au passage :

```
nano /etc/systemd/system/carbon-cache.service
```

Entrer la commande suivante :

```
[Service]
Type=oneshot
RemainAfterExit=yes
ExecStart=/opt/graphite/bin/carbon-cache.py start
ExecStop=/opt/graphite/bin/carbon-cache.py stop
```

On active/configure et démarre le service, `systemctl enable carbon-cache` `chkconfig carbon-cache on` `systemctl start carbon-cache`

on peut tester que notre service fonctionne bien avec la commande suivante :

```
systemctl status carbon-cache
```

## Configuration de Graphite

On va commencer par faire des changements dans le fichier de configuration :

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

Appliquer les changements suivants :

```
#SECRET_KEY = 'UNSAFE_DEFAULT'
à changer avec vos propres caractères
SECRET_KEY = 'sfzadzbfzsmkfk546eDAafhumizfyfz87463zejkm46qaekyeFHPJadjofzopz'
Changer la timezone comme vous le souhaitez
TIME_ZONE = 'Europe/Paris'
```

Nous initialisons ensuite la base de données :

```
django-admin.py syncdb --pythonpath /opt/graphite/webapp --settings graphite.settings
```

si il vous demande de créer un superuser django, choisissez oui ça permettra de sauvegarder les graphes. Pour finir on met Apache comme propriétaire du dossier storage :

```
chown -R apache:apache /opt/graphite/storage
```

## Configuration d'Apache

On va changer le nom du serveur et son port d'écoute :

```
nano /etc/httpd/conf/httpd.conf

Listen 80
à changer avec :
Listen 81

#ServerName www.example.com:80
à changer avec :
ServerName graphite
```

On passe maintenant à la modification du vhost :

```
nano /etc/httpd/conf.d/graphite-vhost.conf
```

On change tout d'abord le port du vhost et ajouter 3 lignes en en-tête :

```
<VirtualHost *:81>
  Header set Access-Control-Allow-Origin "*"
  Header set Access-Control-Allow-Methods "GET, OPTIONS"
  Header set Access-Control-Allow-Headers "origin, authorization, accept"
```

Descendez ensuite à la section directory et changez comme suis :

```
<Directory /opt/graphite/conf/>
  Order deny,allow
  Allow from all
</Directory>

Pour RHEL 6

<Directory /opt/graphite/>
  Options All
  AllowOverride All
</Directory>

Pour RHEL 7

<Directory /opt/graphite/>
  Options All
  AllowOverride All
  Require all granted
</Directory>
```

Pour finir on active apache au démarrage :

```
chkconfig httpd on
service httpd start;sleep 15; service httpd restart
```

## Installation de NetApp Harvest

Si vous avez une baie netapp à monitorer vous devez utiliser netapp harvest qui va ce changer d'envoyer les données graphite. Pour cette partie de la documentation nous allons partir du postula que vous avez placer le .zip de netappharvest dans /tmp.

Commençons par le décompresser :

```
unzip /tmp/netapp-harvest_1.4.1.zip
```

Ensuite nous installons les prérequis :

```
sudo yum install perl-JSON perl-libwww-perl perl-XML-Parser perl-Net-SSLeay perl-Excel-Writer-XLSX
```

et si vous êtes sous RHEL 7 :

```
sudo yum install perl-LWP-Protocol-https
```

Puis on installe le paquet en fonction de la version :

```
yum install -y /tmp/netapp-harvest-1.4.1-1.noarch.rpm
```

On extrait les composants perl :

```
unzip -j netapp-manageability-sdk-5.7.zip netapp-manageability-sdk-5.7/lib/perl/NetApp/* -d /opt/netapp
```

Pour tester l'Installation des deux logiciel (harvest-manager et harvest-worker) nous exécutons les commandes suivantes (vous devriez voir le guide d'utilisation apparaître).

```
/opt/netapp-harvest/netapp-worker  
/opt/netapp-harvest/netapp-manager
```

On active le démarrage automatique :

```
systemctl daemon-reload  
sudo systemctl enable netapp-harvest
```

Configuration de NetApp-harvest :

```
cp /opt/netapp-harvest/netapp-harvest.conf.example /opt/netapp-harvest/netapp-harvest.conf
```

Ne démarrer pas netapp-harvest sans avoir configuré la rétention de graphite. Une fois cela fait vous pouvez démarrer ce service :

```
sudo systemctl start netapp-harvest
```

## Configuration de la baie netapp

!/\ valable pour version de ontap en version 8.3 et supérieur, pour les versions précédente veuillez vous référer à la documentation officielle de netapp

Une fois connecté à votre baie netapp nous allons commencer par créer un rôle :

```
security login role create -role netapp-harvest-role -access readonly -cmddirname "version"  
security login role create -role netapp-harvest-role -access readonly -cmddirname "cluster identity sh
```

```
security login role create -role netapp-harvest-role -access readonly -cmddirname "cluster show"
security login role create -role netapp-harvest-role -access readonly -cmddirname "system node show"
security login role create -role netapp-harvest-role -access readonly -cmddirname "statistics"
security login role create -role netapp-harvest-role -access readonly -cmddirname "lun show"
security login role create -role netapp-harvest-role -access readonly -cmddirname "network interface s
security login role create -role netapp-harvest-role -access readonly -cmddirname "qos workload show"
```

Nous créons ensuite un certificat SSL pour la connexion :

1) Sur votre serveur en mode root faite la commande suivante :

```
cd /opt/netapp-harvest/cert
openssl req -x509 -nodes -days 3650 -newkey rsa:1024 -keyout netapp-harvest.key -out netapp-harvest.pe
```

Maintenant installons le certificat sur le serveur :

```
#!/\ remplacer cluster par le nom de votre cluster
security certificate install -type client-ca -vserver cluster
```

Ensuite copier le contenu du fichier .pem créé sur le serveur avec harvest exemple :

```
cluster::> security certificate install -type client-ca -vserver cluster
Please enter Certificate: Press <Enter> when done
-----BEGIN CERTIFICATE-----
MIICChDCCAe2gAwIBAgIJAKgurBmDXc3uMA0GCSqGSIb3DQEBBQUAMFsx CzAJBgNV
BAYTAk5MMRUwEwYDVQQHDAxEZWZhdWx0IENpdHkxHDAaBgNVBAoME0RlZmFlbHhQg
Q29tcGFueSBMdGQxZzAVBgNVBAMMDm5ldGFwcC1oYXJ2ZXN0MB4XDTE1MDYyNjEw
MTk1N1loXDTI1MDYyMzEwMTk1N1owWzELMAkGA1UEBhMCTkwxFtATBgNVBACMDERl
ZmFlbHhQgQ210eTEcMBoGA1UECgwTRGVmYXVsdCBDb21wYW55IEEx0ZDEXMBUGA1UE
AwwObmV0YXBwLWhhcnZlc3QwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMyq
Qq6qXRW7czWRNHYMfmlZjpr0FV/VmOv0Brt9Ij7+tHYb+CcIKVyj/gv0RM8DGJ5L
X7VrdnnpINAu6tghBS6YOG2Nr1h9CRUnBR91Hm2/DPKA7C0cNjg6EHuJkYLOVF21
nmRpdAXDURBfw89v1YrZz7uc6LBqGX8SRqi0y0OvAgMBAAGjUDBOMB0GA1UdDgQW
BBTOMM2pC8HH0aK9ZRGw5OxOqcV7RDAfBgNVHSMEGDAWgBTOMM2pC8HH0aK9ZRGw
5OxOqcV7RDAMBgNVHRMEBTADAQH/MA0GCSqGSIb3DQEBBQUAA4GBAFrg5HjXtZ8q
YkRcnCyekvdtFT1a18FyWjDUkRtldySyRgsdtwcF6BoYiVvEmjPVX2QR8n6u8G/R
Ii+6Mwt+ODwPTvzZX6k92ni3yDr0Ffghjj9V5+UZEk8aGHPnD4kpt/sAnJf3gbzO
WswIMiWH6mYaYLnkGDAze9UuXZcEuW4E
-----END CERTIFICATE-----
```

Maintenant nous allons autoriser la connection des clients SSL

```
#!/\ remplacer clustername par le nom de votre cluster
security ssl modify -client-enabled true -vserver clustername
```

Pour la création d'utilisateur pour l'api uniquement :

```
security login create -user-or-group-name netapp-harvest -application ontapi -role netapp-harvest-role
```

On change le propriétaire de netapp-harvest :

```
chown netapp-harvest:netapp-harvest /opt/netapp-harvest/netapp-harvest.conf
```

On configure la connection par ssl

```
auth_type          = ssl_cert
ssl_cert           = cert/netapp-harvest.pem
ssl_key            = cert/netapp-harvest.key
```

Ensuite déclarer le cluster :

```
[cluster_1]
hostname = url.fr
group = netapp-harvest
```

Pour finir lancer le poller si vous avez correctement configuré graphite

```
./opt/netapp-harvest/netapp-manager -start
```

## Paramétrage de la rétention

Nous allons commencer par éditer le fichier de configuration :

```
nano /opt/graphite/conf/storage-schemas.conf
```

Commenter l'ensemble du fichier et remplacer le par le contenu suivant :

```
# Schema definitions for Whisper files. Entries are scanned in order,
# and first match wins. This file is scanned for changes every 60 seconds.
#
# [name]
# pattern = regex
# retentions = timePerPoint:timeToStore, timePerPoint:timeToStore, ...

# Carbon's internal metrics. This entry should match what is specified in
# CARBON_METRIC_PREFIX and CARBON_METRIC_INTERVAL settings
#[carbon]
#pattern = ^carbon\.
#retentions = 60:90d

#[default_1min_for_1day]
#pattern = .*
#retentions = 60s:1d
[netapp_perf]
pattern = ^netapp(\.poller)?\.perf7?\.
retentions = 1m:35d,5m:100d,15m:395d,1h:5y
[netapp_capacity]
pattern = ^netapp(\.poller)?\.capacity\.
retentions = 15m:100d,1d:5y
```

## Interfacage avec Grafana

Voici comment ajouter la source de données à Grafana, ensuite libre à vous de créer vos dashboard

## Troubleshooting

Si lors de l'initialisation de la base de données vous avez un problème de caractères non-ASCII dans un fichier .py ajouter **"# coding=utf-8"** en en-tête du fichier. J'ai eu le cas pour ce fichier : `*/opt/graphite/webapp/graphite/local_settings.py`

En cas de fichier de whisper corrompu à cause d'un manque d'espace disque vous pouvez exécuter la commande suivante :

```
grep 'Error writing to ' /opt/graphite/storage/log/carbon-cache/carbon-cache-a/console.log |awk '{print $2}'
How to detect that?

find /opt/graphite/storage/whisper/ -type f -empty

How to fix that?
```

```
find /opt/graphite/storage/whisper/ -type f -empty -delete
```

## Source

Lien de la documentation d'Installation et de configuration \*

<https://community.netapp.com/t5/OnCommand-Storage-Management-Software-Articles-and-Resources/How-to-install-NetApp-Harvest-1.4.1>

Création d'utilisateur : \* <http://blog.asquareb.com/blog/2014/11/19/adding-users-to-graphite/>

Installation de netapp-harvest : \* [NetAppHarvest/NetAppHarvestIAG\\_1.4.1.pdf](#)

Autre

- <https://github.com/graphite-project/carbon/issues/327>