SI5 – Administration des Réseaux
Monitoring, Managing, and Troubleshooting Computer Networks

# TP : Monitoring et Supervision Réseau

## 1 Monitoring réseau avec SNMP

Nous allons maintenant mettre en oeuvre SNMP, et des logiciels d'interrogation automatique pour le monitoring d'un réseau.

Les instructions données dans cette section sont valables pour une distribution GNU/Linux Ubuntu 17.04. Merci de les adapter pour votre système de choix.

### 1.1 Surveillance des interfaces réseau avec MRTG

Pour cette partie du laboratoire, vous aurez besoin de deux machines (soit deux machines virtuelles, soit une machine virtuelle et votre machine physique). De préférence, utilisez votre machine virtuelle comme routeur, dans laquelle vous installerez un *agent* SNMP. Sur une autre machine, vous ferez tourner le *manager* SNMP.

#### 1.1.1 Installation et configuration de l'agent SNMP

Installez l'agent SNMP sur la machine virtuelle :

```
sudo apt-get install snmpd
```

Il faut ensuite configurer l'agent SNMP pour qu'il réponde aux requêtes de MRTG (qui tournera sur le manager). Pour cela, ouvrez le fichier `/etc/snmp/snmpd.conf`. Vous devez modifier la section `AGENT BEHAVIOUR` de manière appropriée pour que l'agent écoute sur toutes les interfaces.

**Question #1 :**
Quelle modification avez-vous apporté ?

Profitez-en aussi pour modifier les informations de votre système. Spécifiez-les dans les champs `SysLocation` et `SysContact`. Ensuite, regardez la section `ACCESS CONTROL`.

**Question #2 :**
Expliquez les instructions `view` et `rocommunity` inclues dans le fichier.
(Vous trouverez la documentation complète des configurations possible pour l'agent SNMP dans la cinquième section des man pages (→ `man snmpd.conf`).)

Supprimez les restrictions de vues et d'accès. Autrement dit, donnez l'accès en lecture à la MIB entière pour tous les managers.

**Question #3 :**
Quelle commande avez-vous utilisé ?

Enregistrez vos changements dans `/etc/snmp/snmpd.conf`, et relancez le daemon SNMP :

```
sudo systemctl restart snmpd.service
```

### 1.1.2 Installation de MRTG et surveillance de l'agent

Maintenant, nous allons installer MRTG sur la machine manager :

```
sudo apt-get install snmp mrtg
```

Si l'installateur vous demande quelque chose, répondez *Non*. Dans un premier temps, vérifiez que votre configuration de l'agent SNMP fonctionne. Appelez `cfgmaker` sur votre agent. Noubliez pas de changer l'adresse IP par celle utilisée par votre agent SNMP (la machine virtuelle).

```
cfgmaker public@192.168.96.135 2>&1 | less -S
```

Le résultat doit lister toutes les interfaces installée dans votre machine virtuelle agent. Si ce n'est pas le cas, il y a un problème dans votre configuration SNMP [1]. Nous allons utiliser MRTG pour générer une page web contenant les informations sur l'agent SNMP. D'abord, installez Apache

```
sudo apt-get install apache2
```

Vous pouvez vérifier que l'installation fonctionne correctement en pointant votre navigateur vers l'adresse http://localhost/.

Créez le fichier `/etc/apache2/sites-available/mrtg.conf` avec la configuration suivante :

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    Alias /mrtg/ /var/www/mrtg/
    <Directory /var/www/mrtg>
        Order allow,deny
        Allow from all
        Require all granted
    </Directory>
</VirtualHost>
```

1. Si vous avez choisi d'être téméraire et de ne pas respecter les instructions en installant l'agent SNMP directement sur votre machine physique, connectée avec son interface WiFi, vous allez rencontrer le problème suivant. MRTG a besoin de connaître la valeur de la vitesse maximale de l'interface pour tracer le débit en fonction du temps. En WiFi, cette valeur varie et n'est pas bien reportée dans SNMP. Vous devez mettre une valeur par défaut, par exemple en spécifiant `--zero-speed=100000000` dans l'appel à `cfgmaker`.

Ensuite, vous créez le répertoire `/var/www/mrtg/`, qui sera utilisé par MRTG pour générer ses données. Vous devez aussi désactiver le site par défaut d'Apache, et activer la configuration MRTG. Finalement, recharger la configuration d'Apache pour que les changements prennent effet.

```
sudo mkdir /var/www/mrtg
sudo a2dissite 000-default
sudo a2ensite mrtg
sudo systemctl reload apache2
```

Vous pouvez vérifier que votre configuration fonctionne en pointant votre navigateur vers l'adresse http://localhost/mrtg/. Vous allez arriver sur une page qui liste les fichiers (pour l'instant, il n'y en a pas) dans votre répertoire `mrtg`.

Générez la configuration pour le daemon mrtg avec `cfgmaker` (n'oubliez pas de changer l'adresse IP !), et lancez MRTG

```
sudo cfgmaker public@192.168.96.135 --output=/etc/mrtg.cfg
LANG=C sudo mrtg /etc/mrtg.cfg
```

Consultez votre liste de fichiers via Apache (adresse : http://localhost/mrtg/. Ouvrez la page web (le fichier `.html`).

**Question #4 :**
Que voyez-vous ?

Vous allez maintenant automatiser la génération des graphes. Pour ce, installez un crontab :

```
sudo crontab -e
```

et ajoutez la ligne

```
* * * * * LANG=C sudo mrtg /etc/mrtg.cfg > /var/log/mrtg-cron.log
```

qui va générer les graphes toutes les minutes. Enregistrez et fermez le fichier cron.

Maintenant, nous allons générer du trafic entre le manager et le serveur. Sur les deux machines, installez `iperf`.

```
sudo apt-get install iperf
```

Sur le manager, lancez le serveur iperf avec la commande

```
iperf -u -s
```

Sur l'agent, créez le script `test.sh` avec le contenu

```
#!/bin/bash
if [ $# -ne 2 ]; then
  echo "USAGE: $0 server time";
  echo "  server - IP address of the server to connect to";
  echo "  time - number of seconds to run the experiment for";
```

```
  exit
fi
SRV=$1
T=$2
while [ $T -gt 0 ];
do
  xp=0
  if [ $T -gt 10 ]; then
    xp=10
    T=$(expr $T - 10)
  else
    xp=$T
    T=0
  fi
  bw=$RANDOM
  let "bw %= 1000"
  mult=$RANDOM
  let "mult %= 1000"
  bw=$(expr $bw \* $mult)
  echo "iperf -u -c $SRV -b $bw -t $xp"
  iperf -u -c $SRV -b $bw -t $xp
done;
echo "Finished."
```

Rendez-le exécutable, et lancez-le. (N'oubliez pas de changer l'adresse IP par celle du manager SNMP.)

```
chmod +x test.sh
./test.sh 192.168.96.134 300
```

Nous générons maintenant du trafic sur l'interface de l'agent, pendant 300 secondes. Laissez-le test tourner, **et revenez répondre à la question dans 5 minutes**.

**Question #5 :**
Insérez et expliquez **un** des graphes obtenus avec MRTG dans votre rapport.

## 1.2 collectd, graphite, et Graphana

### 1.2.1 Installations sur le manager

Comme nous venons de le voir, MRTG est un outil facile à mettre en oeuvre. Mais, il n'est pas simple à personnaliser. Maintenant, nous allons nous focaliser sur la surveillance de serveurs Linux. Sur le manager, nous allons installer une base de données,*graphite*, et un outil de visualisation, *Grafana*. Nous allons aussi installer un serveur web Apache sur l'agent, et le configurer pour qu'il envoie des informations système (càd. des informations d'Ubuntu), ainsi qu'au niveau applicatif (Apache).

Sur le manager, installez graphite :

```
sudo apt-get install libapache2-mod-wsgi graphite-web graphite-carbon
```

Répondez “non” si graphite-carbon vous demande si vous voulez supprimer la base de données.

Changez la valeur de CARBON_CACHE_ENABLED sur true dans /etc/default/graphite-carbon.

Ensuite, copiez le fichier storage-aggregation.conf vers le dossier de configuration, et démarrez le service.

```
sudo cp /usr/share/doc/graphite-carbon/examples/storage-aggregation.
    conf.example /etc/carbon/storage-aggregation.conf
sudo systemctl restart carbon-cache.service
```

Maintenant, nous allons configurer graphite-web. Ouvrez le fichier /etc/graphite/local_settings.py. Décommentez la ligne SECRET_KEY, et mettez-y une chaîne de caractères aléatoire. De plus, modifiez le fuseau horaire Par exemple :

```
SECRET_KEY = 'yphwR5JcuvohZKQCkmCtqlmVCe8hHFgEXF4yCi6wEOFOLoanACyBiq'
TIME_ZONE = 'Europe/Paris'
```

Ensuite, créez la base de données graphite :

```
sudo graphite-manage syncdb
```

Si on vous demande de créer un utilisateur, dites “oui”. Par exemple : graphite/graphite (utilisateur/-mot de passe). Ensuite, changez le propriétaire de la base de données vers l'utilisateur _graphite :

```
sudo chown _graphite:_graphite /var/lib/graphite/graphite.db
```

**N'oubliez pas de remonter sur la Question #5.**

Configurez Apache pour avoir un site graphite. D'abord ajoutez le port 81 en écoute sur Apache. Pour ce, ouvrez /etc/apache2/ports.conf, et ajoutez

```
Listen 81
```

Ensuite, copiez la configuration initiale du vhost Apache pour graphite :

```
sudo cp /usr/share/graphite-web/apache2-graphite.conf /etc/apache2/
    sites-available/graphite.conf
```

Changez le port d'écoute pour celui-ci vers le port 81, autrement dit, changez l'entête du fichier par

```
<VirtualHost *:81>
```

et activez la configuration, et rechargez Apache :

```
sudo a2ensite graphite
sudo systemctl reload apache2
```

Vous pouvez pointez votre navigateur sur http://localhost:81/. Vous devez voir une interface graphique apparaître.

Maintenant, nous allons installer Grafana. Malheureusement, celui-ci n'est pas dans les dépôts officiels, mais nous allons télécharger le paquet depuis le site officiel. Pour cela :

```
wget https://s3-us-west-2.amazonaws.com/grafana-releases/release/
    grafana_4.5.2_amd64.deb
sudo apt-get install -y adduser libfontconfig
sudo dpkg -i grafana_4.5.2_amd64.deb
```

Démarrez le service avec

```
sudo systemctl daemon-reload
sudo systemctl start grafana-server
```

Vous pouvez vérifier que l'installation a fonctionner en pointant votre navigateur sur http://localhost:3000/. Nous allons "jouer" avec un peu plus tard.

### 1.2.2 Installations sur l'agent

Sur l'agent, nous allons installer le daemon `collectd` afin qu'il envoie les données à la base Graphite. Il faut le faire sur chaque serveur (mais, il n'y en a qu'un ici).

```
sudo apt-get install collectd collectd-utils
```

Editez la configuration de `collectd` via le fichier `/etc/collectd/collectd.conf`. Changez le *hostname* par le nom de la machine. Ensuite, vous pouvez activer et désactiver les plugins en commentant ou décommentant les lignes `LoadPlugin`. Activez (au moins) les plugins :

- CPU,
- Entropie,
- Interface

Il faut aussi décommenter leurs configurations respectives plus bas.

Activez également le plugin *write-graphite*, et modifiez sa configuration pour y mettre l'adresse IP du manager (ici : 192.168.96.137)

```
<Plugin write_graphite>
    <Carbon>
        Host "192.168.96.137"
        Port "2003"
        #Prefix ""
        #Postfix ""
        #StoreRates false
        #AlwaysAppendDS false
        #EscapeCharacter "_"
    </Carbon>
</Plugin>
```

et redémarrez le service

```
sudo systemctl status collectd
```

Vous voyez maintenant apparaître l'agent sur l'interface web de graphite (rappel : http://localhost:81/ sur le manager).

### 1.2.3 Grafite

Générez de la charge CPU sur l'agent avec cette implémentation itérative des *tours de Hanoï*. Créez le fichier `hanoi.sh` avec le contenu ci-dessous :

```
#!/bin/bash
disk=50
for (( x=1; x < (1 << $disk ); x++ )) ; do
    i=$((($x & $x - 1 ) % 3))
    j=$(((($x | $x - 1 ) + 1 ) % 3))
done
echo "Finished."
```

et rendez le éxécutable (`chmod +x hanoi.sh`). En parallèle, relancez le script `test.sh`, comme pour MRTG (détaillé à la Section 1.1.2), pour charger l'interface réseau.

**Question #6 :**
Regardez les métriques dans Grafite. Que constatez-vous par rapport à ce que vous aviez plus tôt avec MRTG ? (Nombre de métriques, granularité des données, fréquence de rafraîchissement, ...)

### 1.2.4 Grafana

Connectez-vous maintenant à Grafana. Le nom d'utilisateur/mot de passe est admin/admin.

Ajoutez comme source de données graphite, sur le serveur local (http://localhost:81/, en proxy).

Maintenant, créez un dashboard. Il s'agit d'une collection de figures/graphes/métriques/...Créez 3 graphes :

- un graphe *entropie* ; (regardez sa définition sur https://collectd.org/faq.shtml)
- un graphe CPU, toutes les valeurs ;
- un graphe pour l'interface réseau, mettez le trafic entrant en positif, et le trafic entrant en négatif (pour cela, vous devez faire une transformation `scale(-1)`.

Pensez à sauvegarder votre dashboard régulièrement !

**Question #7 :**
Mettez une capture d'écran de votre dashboard dans le rapport

Installez Apache sur l'agent :

```
sudo apt-get install apache2
```

Pour faire en sorte qu'Apache retourne ses statistiques, d'abord activez le module de stats :

```
sudo a2enmod status
```

ensuite vérifier que cette information est disponible localement en pointant le navigateur du manager vers http://localhost/server-status/. Si ce n'est pas le cas, modifiez la configuration du module `/etc/apache2/mods-available/status.conf` pour que ça soit le cas, et rechargez la configuration.

On veut maintenant traiter ces données avec Grafana. Pour cela, assurez-vous que collectd transmette bien les données Apache vers Grafite. Vous devez activer le module dans `/etc/collectd/collectd.conf`, ainsi que vérifier sa configuration. Redémarrez ensuite collectd :

```
sudo systemctl restart collectd
```

Vous pouvez vérifier que la connexion fonctionne en regardant si les données d'Apache sont apparues dans l'interface web de Grafite.

Ensuite, revenez sur Grafana, et créer un graphe dans votre dashboard avec le nombre de requêtes sur le serveur Apache, et le nombre de connexions par seconde. Lancez un test de résistance sur Apache. Depuis le l'agent, faites

```
ab -n 300000 http://localhost/
```

et, sur l'agent faites

```
ab -n 300000 http://192.168.96.136/
```

**Question #8 :**
Lorsque le test est fini, faites une capture d'écran de votre dashboard Grafana et incluez-là dans votre rapport.