M3105 : Supervision Réseau et Application

October 17, 2022

Vous devez envoyer votre rapport électronique au format PDF par mail à l'enseignant en fin de séance.

1 Création de l'environnement

- Créer deux VMs Debian, une sur chaque machine physique de la paillasse. La première servira de poste d'administration (machine admin) depuis lequel vous lancerez les tests et sur laquelle tournera les outils de supervision et la seconde de serveur d'applications.
- Editer le fichier /etc/hostname sur chaque machine. Changez le nom en *admin* pour la première et *serveur* pour la seconde. Puis rebooter les 2 machines et vérifier que le changement s'est bien fait.

2 SNMP

2.1 Installation et configuration SNMP du serveur d'applications

Installer le service SNMP (l'agent snmpd) sur la machine *serveur* :

```
apt install snmpd
```

Il faut ensuite éditer le fichier de configuration de snmp (/etc/snmp/snmpd.conf) car, de base l'agent SNMP ne répond qu'aux requêtes locales, c'est-à-dire accepte de 127.0.0.1 seulement. - Voir figure 1. Modifier de manière à ce qu'il réponde à tout le monde (la commande à utiliser est déjà dans le fichier, il suffit de la dé-commenter!), et donc à la machine admin. Modifier aussi **sysLocation** et **sysContact**.

```
###############################################################################
#
#  AGENT BEHAVIOUR
#

#  Listen for connections from the local system only
agentAddress  udp:127.0.0.1:161  # Adresse sur laquelle écoute l'agent
#  Listen for connections on all interfaces (both IPv4 *and* IPv6)
#agentAddress udp:161,udp6:[::1]:161


###############################################################################
```

Figure 1: Contrôle accès réseau

On redémarre le serveur SNMP:

```
systemctl restart snmpd
```

2.2 Installation et configuration SNMP du manager

On vérifie que le serveur fonctionne avec la commande snmpwalk depuis la machine **admin** qui permet de parcourir l'arbre SNMP, après avoir installé les outils snmp:

```
[...]# apt install snmp
```

On peut maintenant parcourir l'arbre SNMP :

```
snmpwalk -Os -v 2c -c public IP_serveur
```

Il reste un dernier problème : les OID sont numériques

```
iso.3.6.1.2.1.1.1.0 = STRING: "Linux M3105d 4.9.0-6-amd64 #1 SMP Debian 4.9.82-1+deb9u3 (2018-03-02) x86
```

Pour avoir une version textuelle, il faut modifier le fichier snmp du manager (et non du serveur!), /etc/snmp/snmp.conf, et simplement commenter la ligne avec mib. Il faut aussi installer le paquet qui télécharge les MIBs.

```
[...]# apt install snmp-mibs-downloader
```

Vous devriez maintenant obtenir des OID en mode textuel :

```
sysDescr.0 = STRING: Linux M3105d 4.9.0-6-amd64 #1 SMP Debian 4.9.82-1+deb9u3 (2018-03-02) x86_64
```

Quel est la partie de l'arbre visible SNMP?
On va donner le droit de lecture complet à l'agent. Pour cela il faut éditer le fichier snmpd.conf sur la machine **serveur** en vous aidant de la Figure 2 et retirer la vue systemonly (-V systemonly).

```
###############################################################################
#
#  ACCESS CONTROL
#

                                #  system + hrSystem groups only
view   systemonly  included   .1.3.6.1.2.1.1
view   systemonly  included   .1.3.6.1.2.1.25.1

                                #  Full access from the local host
#rocommunity public  localhost
                                #  Default access to basic system info
 rocommunity public  default    -V systemonly
# rocommunity = communauté readonly. Communité = mot de passe
# Valeur par défaut est toujours public → à modifier !
# -V systemonly : la personne utilisant ce mot de passe ne verra que le bout de l'abre définit
# par systemonly quelques lignes plus haut
```

Figure 2: Contrôle accès données

Si on refait maintenant un

```
snmpwalk -Os -v 2c -c public IP_serveur
```

on doit voir beaucoup plus d'objets.
Y a-t-il des informations sur le service Apache?
La communauté SNMP est utilisée comme mot de passe et la valeur par défaut est en général toujours "public" pour la communauté lecture seule (rocommunity). Changer le nom de cette communauté en un autre nom et vérifiez que cela fonctionne :
Si on refait maintenant un

```
snmpwalk -Os -v 2c -c nouveau_nom _communaute IP_serveur
```

2.3 Sécurisation et Manager évolué

2.3.1 SNMPv2

Snmp v2 n'est pas très sécurisé. Pour vous en rendre compte, **installez tcpdump** sur la machine **admin**:

```
[...]# apt install tcpdump
```

Puis démarrez le en écoute sur le port UDP numéro 161 (port SNMP), en le faisant écrire les 10 premières trames en ASCII dans le fichier snmp.dmp

```
[...]# tcpdump -c 10 -A -i en1 -n 'udp and port 161' -w snmp.txt &
```

Ensuite relancez la commande snmpwalk et quand elle a fini, lisez le fichier snmp.txt.
Que remarquez-vous, du point de vue sécurité?

2.3.2 SNMPv3

On va passer en SNMPv3. Pour cela il faut tout d'abord arrêter le service snmpd sur la machine serveur :

```
[...]# systemctl stop snmpd
```

Puis on crée l'utilisateur, par exemple m3105, avec la commande suivante (on utilise un mot de passe assez long, le même pour la confidentialité (AES) et l'intégrité (SHA), par exemple Mot2Passe):

```
[...]# apt install libsnmp-dev
[...]# mkdir /snmp
[...]# net-snmp-config --create-snmpv3-user -a SHA -x AES
[...]# cat /snmp/snmpd.conf >> /etc/snmp/snmpd.conf
```

Il faut ensuite recopier le fichier /snmp/snmpd.conf
Puis on redémarre le serveur :

```
[...]# systemctl restart snmpd
```

et on vérifie qu'on peut faire l'interrogation en SNMPv3:

```
[...]# snmpwalk -v 3 -l authPriv -u m3105 -a SHA -A "Mot2Passe" -x AES -X
    "Mot2Passe" IP_Server
```

Si cela fonctionne, on édite le fichier /etc/snmp/snmpd.conf et on commente les parties liées à l'autentification en V2, c'est-à-dire **la communauté**. On fait un nouveau redémarrage:

```
[...]# systemctl restart snmpd
```

et on vérifie que seul SNMPv3 fonctionne.
Il faut faire des copies d'écran qui montrent que snmpwalk fonctionne en V3 et plus en V2, c'est-à-dire que la commande snmpwalk en V2 ne doit plus fonctionner.

3 Netdata

Netdata (`https://www.netdata.cloud/`) est un outil qui s'installe sur le serveur, capture **toutes** les métriques qu'il trouve - système, application, réseau - et les affiche dans un serveur Web.

3.1 Installation

On installe netdata dans le **serveur**

```
[...]# bash <(curl -Ss https://my-netdata.io/kickstart.sh) --dont-wait
```

Depuis la machine **admin**, on interroge le serveur netdata qui tourne sur le port 19999 au travers d'un navigateur Web (`http://IP_serveur:19999`).

3.2 Benchmarking Apache

Sur la machine **serveur**, on crée un gros fichier de 100 Moctets :

```
[...]# dd if=/dev/random of=/var/www/html/large.txt bs=1 count=0 seek=100
    M
```

On lance ab sur le gros puis le petit fichier (qui est simplement index.html) avec les commandes suivantes :

```
ab -n 100000 -c 10 http://IP_serveur/index.html
ab -n 100000 -c 10 http://IP_serveur/large.txt
```

. . . et **on fait des copies d'écrans de Netdata**:

- de la consommation CPU
- du trafic réseau
- du nombre de requêtes GET faites sur le serveur

3.3 Collecte en base de données puis Dashboards customisés avec Grafana

Netdata stocke ses données de manière compacte, mais il ne garde que 30 minutes de données. Nous allons exporter les données dans un outil de management évolué, Prometheus (`https://prometheus.io/`). Prometheus va seulement servir de stockage intermédiaire. Plus précisément il va servir à stocker les séries temporelles des données générées par Netdata. Nous allons ensuite installer Grafana et le connecter à Prométheus pour pouvoir ensuite afficher les données .

3.3.1 Installation Prometheus

```
[...]# apt install prometheus
```

Il faut ensuite éditer le fichier /etc/prometheus/prometheus.yml et ajouter à la partie scraping le job netdata en **mettant la bonne adresse**.
Attention : lors de l'édition du fichier, n'utilisez pas de tabulations, mais uniquement des espaces!!! Sinon, cela va donner une erreur.
Attention : il faut des quotes simples dans les fichiers !!!!

```
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: prometheus

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ['localhost:9090']

  - job_name: netdata

    metrics_path: /api/v1/allmetrics
    params:
      format: [ prometheus ]

    static_configs:
      - targets: ['IP_serveur:19999']
```

Redémarrer ensuite Prometheus et se loger à l'adresse `http://IP_admin:9090/`.
Pour vérifier que cela fonctionne, il suffit de taper netdata dans la partie recherche et les métriques doivent apparaître, comme en Figure 3.

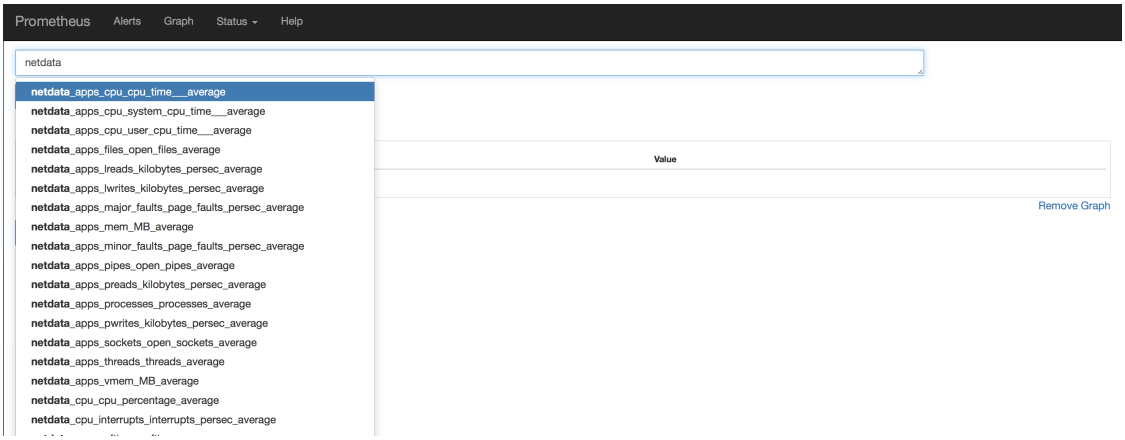

Figure 3: Métrique Netdata dans Prometheus

3.3.2 Installation Grafana

L'installation se fait sur la machine **admin**.
L'installation de Grafana (**version OSS** et pas version entreprise) se fait ensuite en suivant les instructions de la page
`https://grafana.com/docs/installation/debian/#apt-repository`.
Il faut soit faire l'installation en rt, soit en root mais il faut retirer les sudo.

Une fois l'installation terminée, il faut démarrer grafana

```
[...]# systemctl start grafana-server
```

3.3.3 Mise en oeuvre Grafana

- On se loge ensuite sur l'interface de Grafana : `http://IP_admin:3000/` avec les identifiants admin/admin
- On ajoute une source de données et on choisit Prometheus et on met l'adresse de la machine **admin**.
- On crée un Dashboard "Serveur - Netdata" et on crée des graphes. Pour choisir les métriques, se référer à `http://IP_admin:19999/api/v1/allmetrics?format=prometheus&help=yes`
  qui est l'adresse où Netdata exporte les méta-données pour les données que Prométheus récupèrera. On veut 3 graphes :
  - La consommation CPU globale : utilisez le mot clef cpu_cpu pour trouver les métriques intéressantes
  - Le trafic réseau : utilisez le mot clef system_net pour trouver les métriques intéressantes
  - Le nombre de requête du serveur Apache2 : utilisez le mot clef web_log pour trouver les métriques intéressantes

  Cela devrait ressembler (un peu) à la figure 4.

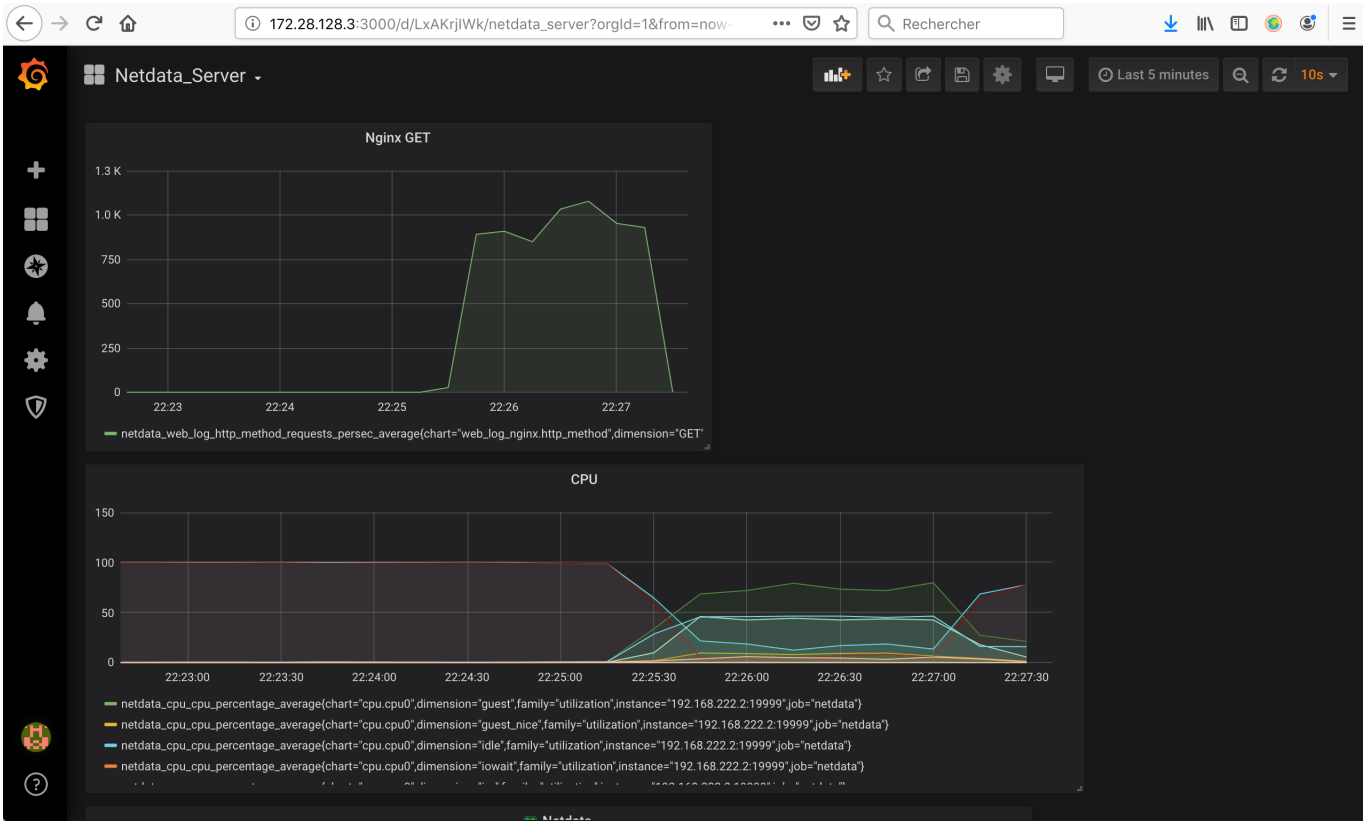

Figure 4: Dashboard Grafana