M3105 : Supervision Réseau et Application SNMP

1. SNMP

a. Installation de SNMP

Pour faire du SNMP il faut installer snmpd sur le serveur que l'on veut superviser et snmp sur l'ordinateur administrateur.

Sur le serveur :

 # apt install snmpd

Sur l'ordinateur de l'administrateur :

 # apt install snmp

Pour l'instant seul le serveur lui-même à accès aux données de SNMP, il faut donc lui dire d'écouter pour toutes les adresses IP et plus seulement local host.

Cela se fait dans le fichier de configuration du serveur (/etc/snmp/snmpd.conf).

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


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

Il faut commenter la ligne en jaune pour qu'elle ne soit plus prise en compte et décommenter la dernière. Aucune adresse IP n'est spécifiée donc toutes les adresses sont acceptées.

Une fois fait et le serveur redémarré, l'admin peut accéder aux données snmp via la commande : « snmpwalk »
Exemple :

 # snmpwalk -Os -v 2c -c public 10.1.0.101

b. Configuration SNMP coté admin

Il reste un dernier problème : les OID sont numériques. Pour avoir une version textuelle et donc plus interprétable qu'une suite de nombre, il faut modifier le fichier snmp du manager, /etc/snm-p/snmp.conf, et simplement commenter la ligne avec mib. Il faut aussi installer le paquet qui télécharge lesMIBs.
apt install snmp-mibs-downloader

```
sysORDescr.1 = STRING: The MIB for Message Processing and Dispatching.
sysORDescr.2 = STRING: The management information definitions for the SNMP U
y Model.
sysORDescr.3 = STRING: The SNMP Management Architecture MIB.
sysORDescr.4 = STRING: The MIB module for SNMPv2 entities
```

On voit une version textuelle en début de ligne.

Pour l'instant la partie de l'arbre visible est très limitée, nous avons que quelques infos sur le serveur. Nous avons uniquement accès à la partie read-only.
Les OID numériques accessibles par la communauté « public », par défaut sont 1.3.6.1.2.1.1 et 1.3.6.1.2.1.25.1.

On va donner le droit de lecture complet à l'agent. Pour cela il faut éditer le fichier snmpd.conf sur la machine serveur et retirer la vue systemonly (-V systemonly)
Maintenant lorsque l'on fait :

 # snmpwalk -Os -v 2c -c public 10.1.102.101

On voit beaucoup plus d'objet.

Si on fait un « grep apache » sur le résultat, on voit des lignes contenants apache.

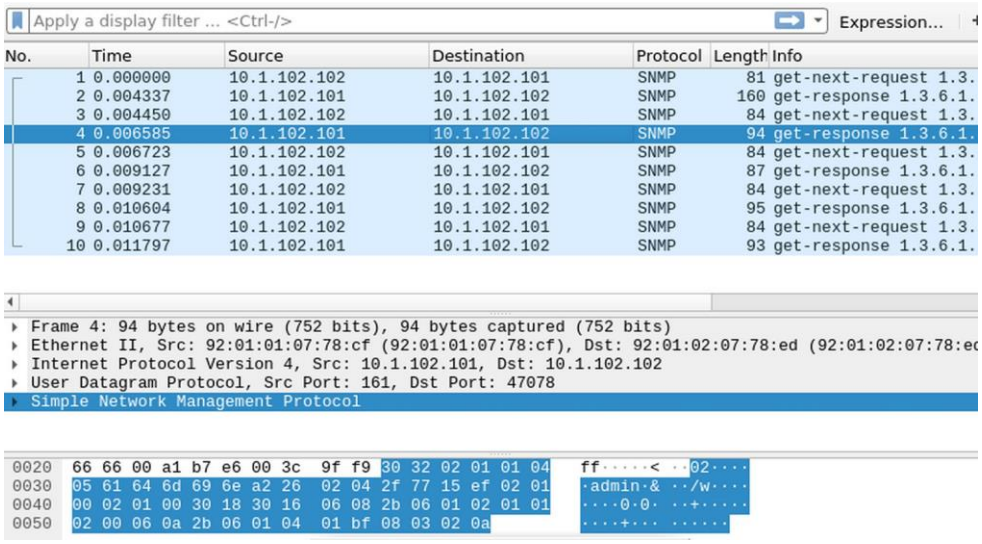
```
root@rt102p102:~# snmpwalk -Os -v 2c -c admin 10.1.102.101 | grep apache
hrSWRunName.945 = STRING: "apache2"
hrSWRunName.949 = STRING: "apache2"
hrSWRunName.950 = STRING: "apache2"
hrSWRunName.951 = STRING: "apache2"
hrSWRunName.952 = STRING: "apache2"
hrSWRunName.953 = STRING: "apache2"
hrSWRunPath.945 = STRING: "/usr/sbin/apache2"
hrSWRunPath.949 = STRING: "/usr/sbin/apache2"
hrSWRunPath.950 = STRING: "/usr/sbin/apache2"
hrSWRunPath.951 = STRING: "/usr/sbin/apache2"
hrSWRunPath.952 = STRING: "/usr/sbin/apache2"
hrSWRunPath.953 = STRING: "/usr/sbin/apache2"
hrSWInstalledName.12 = STRING: "apache2-2.4.38-3+deb10u5"
hrSWInstalledName.13 = STRING: "apache2-bin-2.4.38-3+deb10u5"
hrSWInstalledName.14 = STRING: "apache2-data-2.4.38-3+deb10u5"
hrSWInstalledName.15 = STRING: "apache2-doc-2.4.38-3+deb10u5"
hrSWInstalledName.16 = STRING: "apache2-utils-2.4.38-3+deb10u5"
hrSWInstalledName.418 = STRING: "libapache-poi-java-4.0.1-1"
hrSWInstalledName.419 = STRING: "libapache-pom-java-18-1"
hrSWInstalledName.420 = STRING: "libapache2-mod-php7.3-7.3.29-1~deb10u1"
```

Informations sur le service apache sur le serveur depuis l'admin

c. Les communautés

L'utilisation d'une autre communauté en supprimant le (-V systemonly) permet d'afficher ces informations (non disponibles dans la communauté public) nous l'avons appelée cette communauté « admin ». Sur SNMP 1 et 2 seul le nom de la communauté permet de protéger les accès, donc si quelqu'un de mal intentionné trouve le nom de la communauté même en read-only, il peut trouver des informations sur le système (comme on a vu avec la version d'apache) et pire, si la communauté est en read-write, il peut complètement reconfigurer le serveur et en faire ce qu'il veut.

Les noms de communauté sont en plus communiqués en clair, nous allons le démontrer en utilisant tcpdump. On fait une capture du trafic (pendant qu'on fait des requêtes snmp) coté client par exemple et on l'ouvre avec wireshark et là, on voit le nom de la communauté.



Visualisation d'une trame SNMP avec le nom de la communauté en clair. Ici « admin »

On voit le nom de la communauté dans la capture tcpdump ouverte avec wireshark (dans tous les paquets) ici « admin ».
Donc il suffit de sniffer les paquets pour trouver le nom de la communauté.

d. SNMP v3

Donc pour plus de sécurité on passe à SNMP v3 qui n'a plus de concept de communauté mais d'authentification. Pour ce faire :
On stop le service sur le serveur :

 # systemctl stop snmpd

Puis on installe snmp V3

 # apt install libsnmp-dev

Puis on crée un utilisateur avec :

 # net-snmp-config --create-snmpv3-user -a SHA -x AES

On indique le login et les mots de passe.
Puis on redémarre snmp avec :

#systemctl restart snmpd

Depuis l'admin on peut maintenant se connecter avec une connexion sécurisée.
Il faut commenter les lignes concertants les communautés sur le fichier de conf sur le serveur, pour avoir uniquement de l'authentificaiton sécurisé snmp v3.
La commande snmp v2 ne fonctionne donc plus, à dessein.

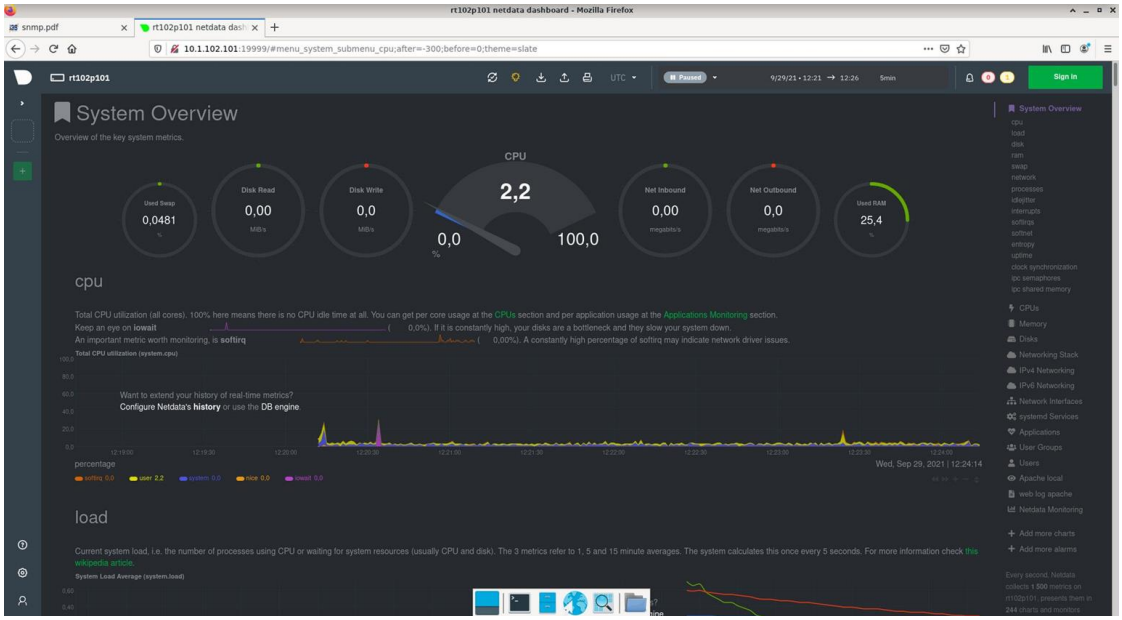
2. NETDATA

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.
Pour ce faire : On installe netdata dans le serveur :

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

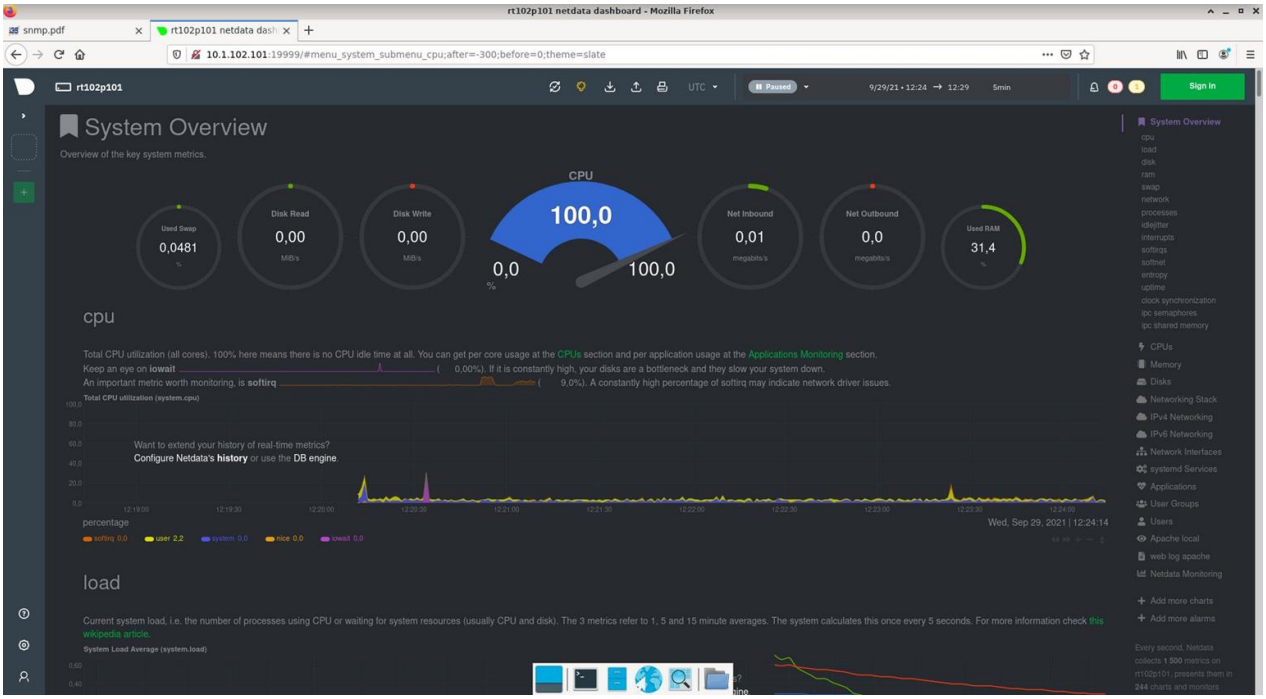
Puis on interroge le serveur netdata qui tourne sur le port 19999 au travers d'un navigateur web (http://IP_serveur:19999)

Donc en l'occurrence pour nous :



Capture d'écran de netdata sans opération particulière

On voit, depuis le client les informations affichées par netdata.
Si l'on demande au serveur de faire plein de requêtes on remarque que le CPU est à 100 %.



Capture d'écran de netdata lorsque le serveur est en train de faire de grosses opérations

3. PROMETHEUS

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.

Une fois prometheus installé sur la machine administrateur grâce à :

 # apt install prometheus

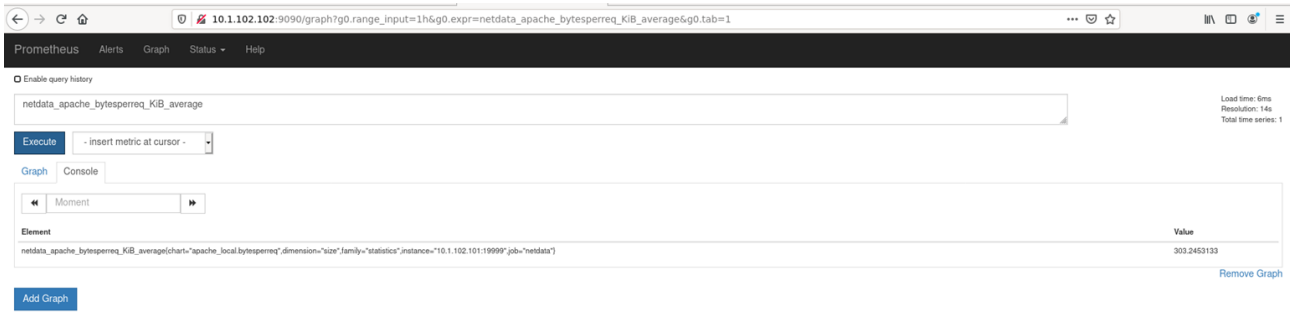
Il faut le configurer de manière à ce qu'il aille récupérer les données chez netdata, sur le serveur. Tel que :

```
- job_name: 'netdata'

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

  static_configs:
    - targets: [ '10.1.102.101:19999' ]
```

Une fois prometheus redémarré, on peut acceder aux données à l'adresse :

http://10.1.102.102:9090/

Pour vérifier que cela fonctionne, il suffit de taper netdata dans la partie recherche et les métriques doivent apparaître, comme ci-dessous :


10.1.102.102:9090/graph?g0.range_input=1h&g0.expr=netdata_apache_bytesperreq_KiB_average&g0.tab=1
Prometheus
Alerts
Graph
Status
Help
Enable query history
netdata_apache_bytesperreq_KiB_average
Load time: 6ms
Resolution: 14s
Total time series: 1
Execute
- insert metric at cursor -
Graph
Console
Moment
Element
Value
netdata_apache_bytesperreq_KiB_average{chart="apache_local.bytesperreq",dimension="size",family="statistics",instance="10.1.102.101:19999",job="netdata"}
303.2453133
Remove Graph
Add Graph

4. GRAFANA

Pour pouvoir exploiter les données de prometheus on va donc utiliser 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.

Une fois installé, on démarre le service :

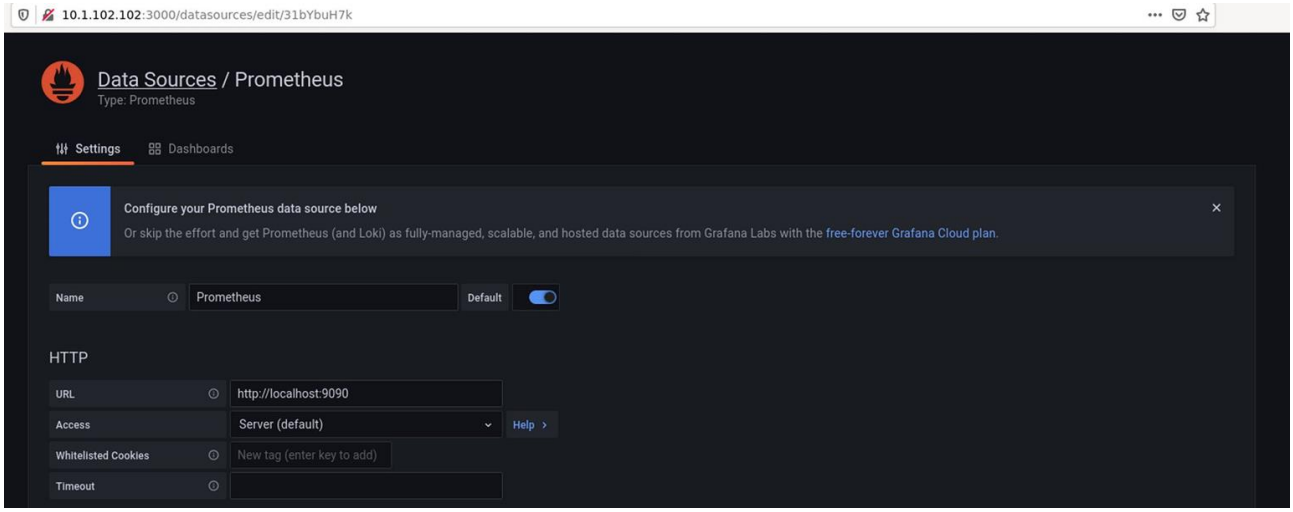
#systemctl start grafana-server

On se loge ensuite sur l'interface de Grafana :

http://10.1.0.102:3000/

On utilise les identifiants admin/admin
On ajoute une source de données et on choisit Prometheus et on met l'adresse de la machine admin

Comme tel :



Capture d'écran de l'interface de configuration des sources de grafana.

Puis, on crée un Dashboard "Serveur - Netdata" et on crée des graphes. Pour choisir les métriques, se référer à :

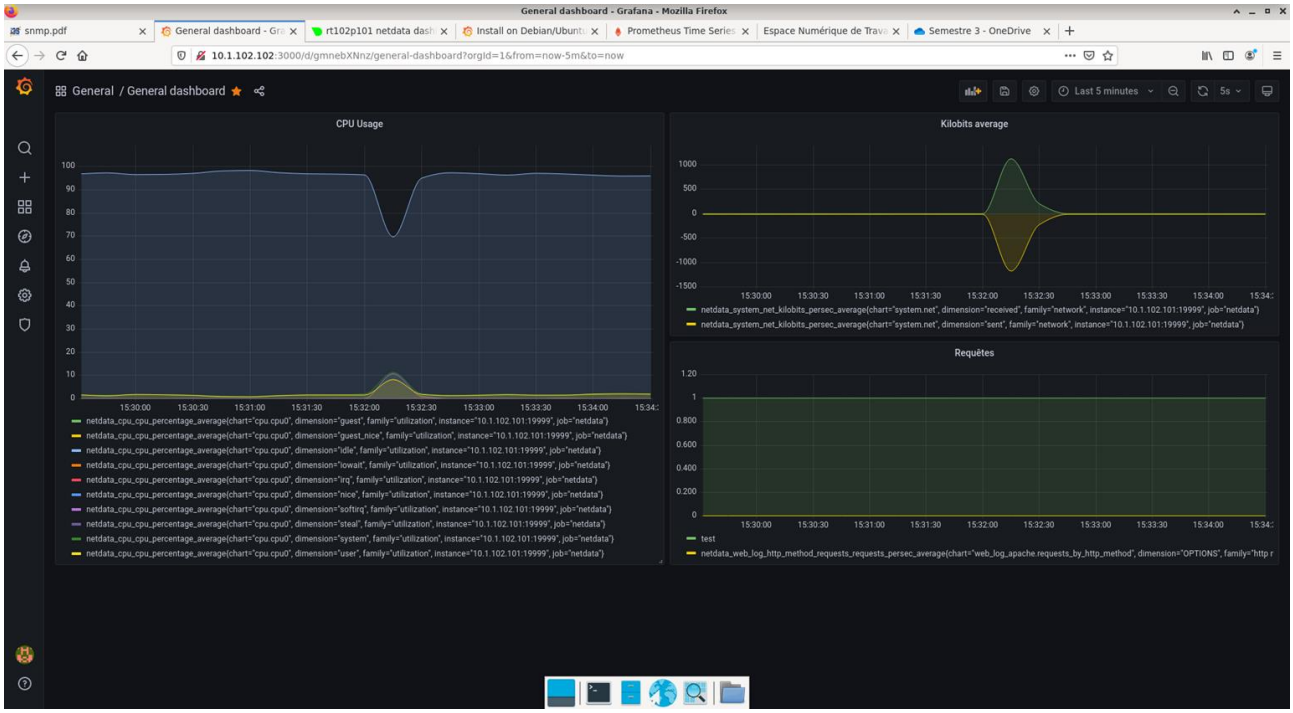
http://IP_admin:19999/api/v1/allmetrics?format=prometheus&help=yes

C'est l'adresse où Netdata exporte les méta-données pour les données que Prométheus récupèrera. On veut 3 graphes :

- La consommation CPU globale : on utilise le mot clef cpu_cpu pour trouver les métriques intéressantes

- Le trafic réseau : on utilise le mot clef system_net pour trouver les métriques intéressantes
- Le nombre de requête du serveur Apache2 : on utilise le mot clef web_log pour trouver les métriques intéressantes

Finalement, on obtient ça :



Capture d'écran de grafana une fois les graphiques définis

Le nom des courbes est un peu barbare, il faudrait donc les modifier dans les paramètres de grafana, par quelque chose de plus parlant.