

Protocole SNMP

1) Introduction



SNMP (Simple Network Management Protocol) est un protocole de communication permettant, aux administrateurs réseaux, de récupérer des informations sur des équipements compatibles SNMP (ordinateurs, serveurs, switches, routeurs, pare-feux matériels, bornes WiFi, imprimantes...), afin de diagnostiquer des problèmes et plus généralement les superviser (en relevant rapidement certaines valeurs comme la RAM, le CPU, etc.). Et oui, énormément d'informations sur les appareils sont stockés et facilement accessibles à distance via SNMP.

Les communications SNMP se font via les ports 161 et 162 en UDP.

Généralement, SNMP se combine avec un logiciel de supervision (comme Centreon, Nagios ou Shinken) afin d'automatiser les requêtes et d'avoir un tableau de bord convivial (et d'autres fonctionnalités plus poussées).

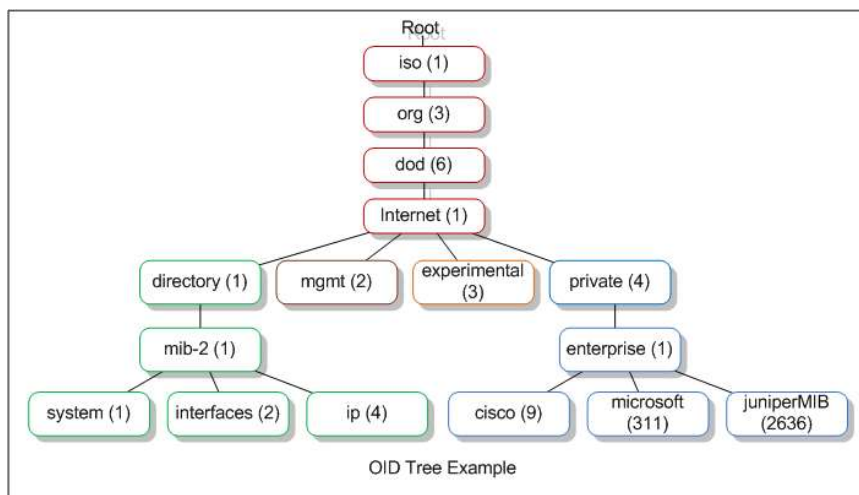
2) Organisation des informations

Les équipements compatibles SNMP, une fois SNMP activé, font tourner un agent SNMP (processus ou daemon, qui répond aux requêtes du réseau). Cet agent fournit un grand nombre d'informations classées en identifiants d'objets (Object Identifiers ou OID).

Un OID porte un nom (issu d'une arborescence spécifique) ou numéro, ainsi qu'une valeur. Par exemple, sur un switch CISCO, l'OID `cpmCPUTotal5minRev` (numéro : `.1.3.6.1.4.1.9.9.109.1.1.1.8`) donne le pourcentage global du processeur occupé au cours des 5 dernière min. Si on fait une requête (depuis un manager SNMP ou serveur SNMP) sur cet OID, on aura par exemple pour valeur retour : 34 (% de CPU utilisé).

L'ensemble des OID forme une MIB (Management Information Base). C'est une sorte de base de données, avec une arborescence, qui permet de se repérer facilement dans les OID (elle les recense et les décrit pour simplifier la vie des administrateurs). On peut s'en passer pour faire des requêtes mais il est tout de même recommandé de s'y référer (par exemple, en téléchargeant les MIB des équipements à superviser, directement via le site web du constructeur) afin de ne pas avoir à mémoriser tous les OID !

Exemple de structure d'une MIB (schéma ci-contre).



Exemple de contenu (aide) d'une MIB :

```
sysUpTime OBJECT-TYPE
    SYNTAX      TimeTicks
    ACCESS      lecture uniquement
    STATUS      obligatoire
    DESCRIPTION
        "The time (in hundredths of a second) since the
         network management portion of the system was last
         re-initialized."
    ::= { system 3 }

sysUpTime OBJECT-TYPE
    Définit l'objet sysUpTime.

SYNTAX TimeTicks
    Objet de type TimeTicks. Les types d'objet sont spécifiés dans le SMI mentionné précédemment.

ACCESS read-only
    Cet objet peut être uniquement lu par SNMP (requête get) ; il ne peut être modifié (requête set).

STATUS mandatory
    Cet objet doit être mis en oeuvre sur n'importe quel agent SNMP.

DESCRIPTION
    Description de l'objet

::= { system 3 }
    L'objet sysUpTime constitue la troisième branche de l'arborescence du groupe d'objets système.
```

3) Sécurité

Il existe plusieurs versions du protocole SNMP. La version 1 est peu sécurisée. La version 2c est la plus courante, améliorée par rapport à v1 mais pas sécurisée. La version 3 est sécurisée (chiffrement).

De plus, pour sécuriser les accès, il est nécessaire de spécifier (sur l'agent SNMP et le serveur SNMP) un nom de communauté (community). Il s'agit d'une sorte de login et mot de passe (afin que tout le monde ne puisse pas requêter sur l'agent). Il est donc important d'utiliser un autre nom que « public »...

Enfin, la communauté peut se voir attribuer des droits en lecture ou écriture. Il est recommandé de sélectionner uniquement le droit de lecture. Sinon, avec le droit en écriture, un utilisateur peut modifier les données du système cible.

4) Techniques de supervision avec SNMP

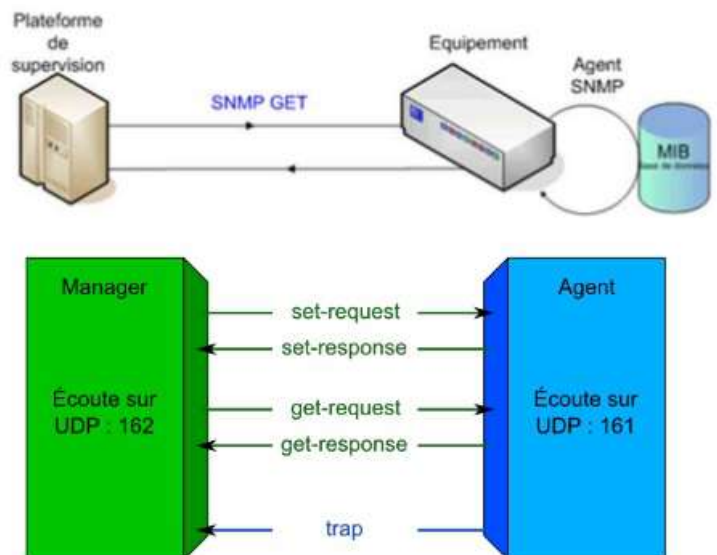
SNMP peut être utilisé de 2 façons : via des requêtes directes ou actives (polling) ou via un mode passif (avec des traps).

Le polling consiste à envoyer une requête à intervalles réguliers pour obtenir une valeur précise.

Les traps consistent à faire de la vérification passive : l'agent SNMP de l'équipement supervisé n'envoie une valeur (trap SNMP) que lors de certains événements (par exemple si un câble est débranché sur un port spécifique).

Pour l'instant, on va se focaliser uniquement sur le polling. Pour le polling, on distingue plusieurs types de requêtes :

1. SNMPGET : permet de récupérer la valeur d'un OID.
2. SNMPWALK : permet de récupérer toutes les valeurs d'un OID de type « nœud », c'est-à-dire contenant d'autres OID. On obtient donc plusieurs réponses. Attention : il ne faut jamais faire un SNMPWALK sur la racine ou un nœud de haut niveau, cela saturerait l'agent SNMP, le réseau et votre poste...
3. SNMPTABLE : permet d'afficher le résultat sous forme d'un tableau (davantage lisible).



On peut également avoir des SNMPSET pour écrire une valeur ou TRAP pour éviter de faire des requêtes GET et attendre une TRAP, alerte, lorsqu'un événement survient. Ou encore GET-NEXT pour interroger l'OID suivant.

5) **DECOUVERTE (phase optionnelle)** Visualisation graphique de la MIB sur Windows et requêtes SNMP

Etape 1 : Installation et activation de l'agent SNMP Windows

Ajout de la fonctionnalité à la liste :

Installation de la fonctionnalité : Menu Démarrer, saisir Gérer les fonctionnalités facultatives, et ajouter la fonctionnalité Protocole SNMP. Une fois installée, la fonctionnalité apparaît dans la liste des fonctionnalités.

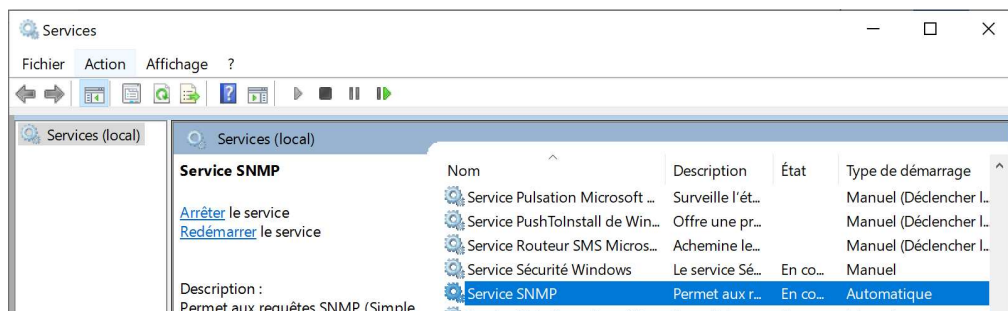


Vérification que le service est en cours d'exécution (running) :

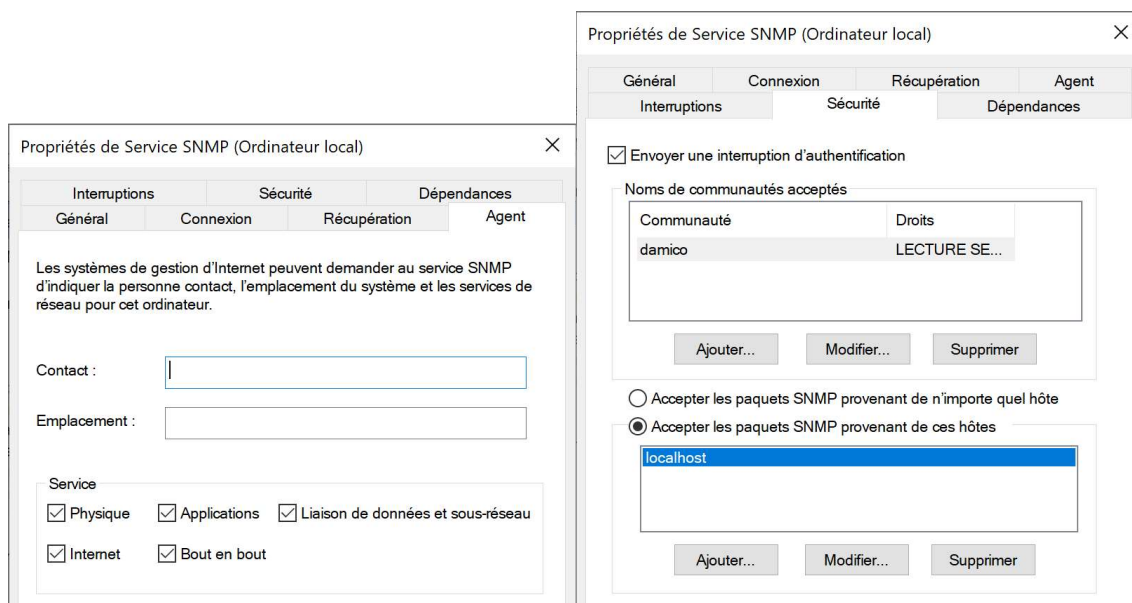
```
PS C:\WINDOWS\system32> get-service -Name snmp*

Status Name DisplayName
-----
Running SNMP Service SNMP
Stopped SNMPTRAP Interruption SNMP
```

Configuration du service SNMP : aller dans Services (via le Menu Démarrer) et cliquer droit sur Service SNMP, puis Propriétés.



Sélectionner tous les services dans l'onglet Agent. Dans l'onglet Sécurité, configurer la communauté et la sécurité en n'acceptant les paquets SNMP que depuis son propre poste (localhost).

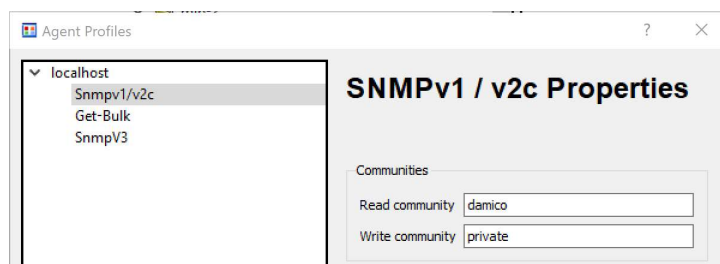


Etape 2 : Installation d'un navigateur de MIB

Avec le logiciel « SnmpB », on peut parcourir la MIB Windows et effectuer des requêtes. Le télécharger et l'installer.

Configurer la communauté : menu Options, Manage Agent Profiles.

- Le nom de la communauté définie sur le ou les agents SNMP, afin de s'identifier (ici, la communauté s'appelle « damico » et elle a été définie en lecture seul sur l'agent ← notre PC Windows)



Paramétrer les propriétés générales :

- Le nom du profil
- Le nom ou l'adresse IP de l'agent SNMP (ici 127.0.0.1 car on va s'interroger soi-même)
- Le numéro du port d'écoute SNMP (port qui recevra nos requêtes SNMP)

General Properties

Profile
Name: localhost

Target SNMP Agent
Agent Address/Name: 127.0.0.1
Agent Port: 161

Timeout and Retries
Retries: 1
Timeout (sec): 3

Supported SNMP Version
☒ SNMPV1
 ☒ SNMPV2
 ☒ SNMPV3

Etape 3 : Parcours de la MIB et requêtes GET, WALK ou TABLE

Dérouler la MIB, sélectionner les OID (donc informations) souhaités, puis cliquer droit sur un OID et choisir : GET (pour avoir la valeur) ou WALK (pour avoir toutes les valeurs même enfants) ou encore TABLE pour afficher les valeurs sous forme d'un tableau.

SnmpB

File Tools Options Help

Tree Modules Editor Discovery Traps Graphs Log

Remote SNMP Agent: localhost

MIB Tree

- iso
 - org
 - dod
 - internet
 - directory
 - mgmt
 - mib-2
 - system
 - sysDescr
 - sysObjectID
 - sysUpTime
 - sysContact
 - sysName (selected)
 - sysLocation
 - sysServices
 - sysORLastChange
 - sysORTable
 - interfaces
 - at
 - ip
 - icmp
 - tcp
 - udp
 - egp
 - transmission

Query Results

```

-----SNMP query started-----
1: sysName.0 SURFACE-GREGORY
-----SNMP query finished-----
Total # of Requests = 1
Total # of Objects = 1
  
```

Node Info

Name:	sysName
Oid:	1.3.6.1.2.1.1.5
Composed Type:	DisplayString
Base Type:	OCTET STRING
Status:	current
Access:	read-write
Kind:	Scalar
SMI Type:	OBJECT-TYPE
Size:	0 .. 255
Module:	SNMPv2-MIB
Description:	An administratively-assigned name for this managed node. By convention, this is the node's fully-qualified domain name. If the name is unknown, the value is the zero-length string.

Autre exemple avec un TABLE VIEW sur les disques de stockage :

Query Results

-----SNMP query started-----
Collecting table objects, please wait ...

Instance	hrStorageIndex	hrStorageType	hrStorageDescr	hrStorageAllocationUnits	hrStorageSize	hrStorageUsed	hrStorageAllocationFailureCount
1	1	hrStorageFixedDisk	C:\ Label: Windows Serial Number fe60f935	4096	62161663	49254333	0
2	2	hrStorageRemovableDisk	D:\	0	0	0	0
3	3	hrStorageVirtualMemory	Virtual Memory	65536	150356	112140	0
4	4	hrStorageRam	Physical Memory	65536	129876	88240	0

-----SNMP query finished-----
Total # of rows = 4

Node Info

Name: hrStorageEntry
Oid: 1.3.6.1.2.1.25.2.3.1
Composed Type:
Base Type:
Status: current
Access: not-accessible
Kind: Row
Index(es): hrStorageIndex
SMI Type: OBJECT-TYPE
Module: HOST-RESOURCES-MIB
Description: A (conceptual) entry for one logical storage area on the host. As an example, an instance of the hrStorageType object might be named hrStorageType.3

6) Exemples d'OID à superviser

Equipement cisco (routeur et switch) :

- hostName (.1.3.6.1.4.1.9.2.1.3) : Hostname de l'équipement
- cpmCPUTotal5minRev (.1.3.6.1.4.1.9.9.109.1.1.1.8) Le pourcentage global du processeur occupé au cours des 5 dernière min.
- cpmCPUTotal1minRev (.1.3.6.1.4.1.9.9.109.1.1.1.7) Le pourcentage global du processeur occupé au cours de la dernière minute
- sysDescr (.1.3.6.1.2.1.1.1) Description textuelle de l'entité, nom complet, identification ...
- sysUpTime (.1.3.6.1.2.1.1.3) Temps écoulé depuis la mise en marche de l'appareil.
- dot1dTpFdbAddress (.1.3.6.1.2.1.17.4.3.1.1) Adresse mac de la Table de diffusion
- dot1dTpFdbPort (.1.3.6.1.2.1.17.4.3.1.2) Port associé à la table de diffusion
- ifOperStatus (.1.3.6.1.2.1.2.2.1.8) L'état actuel de fonctionnement de l'interface (Up/down ...).
- ifSpeed (.1.3.6.1.2.1.2.2.1.5) Une estimation de la bande passante actuelle de l'interface en bits par seconde.
- ifHighSpeed (.1.3.6.1.2.1.31.1.1.15)
- ifNumber (.1.3.6.1.2.1.2.1) le nombre d'interfaces réseaux de l'équipement
- ifOutOctets (.1.3.6.1.2.1.2.2.1.16) le nombre d'octets transmis depuis l'interface
- ifMtu (.1.3.6.1.2.1.2.2.1.4) La taille du plus grand paquet qui peut être envoyé/reçu sur l'interface, spécifiée en octets

- ifLastChange (.1.3.6.1.2.1.2.2.1.9) La valeur depuis le moment où l'interface a été opérationnelle.
- ifInErrors (.1.3.6.1.2.1.2.2.1.14) le nombre de paquets entrant contenant des erreurs.
- ifOutErrors (.1.3.6.1.2.1.2.2.1.20) le nombre de paquets sortant contenant des erreurs.
- atTable (.1.3.6.1.2.1.3.1) La table de translation d'adresses IP et adresses MAC.
- ipInReceives (.1.3.6.1.2.1.4.3) Le nombre total de datagrammes d'entrée reçus de interfaces, y compris ceux reçus par erreur.
- ipRouteTable (.1.3.6.1.2.1.4.21) La table de routage IP de l'équipement
- authAddr (.1.3.6.1.4.1.9.2.1.5) Dernière authentification échouée
- ciscoMemoryPoolUsed (.1.3.6.1.4.1.9.9.48.1.1.1.5) Indique le nombre d'octets à partir du pool de mémoire qui sont actuellement en cours d'utilisation par les applications sur le périphérique géré.
- ciscoMemoryPoolUtilization10Min (.1.3.6.1.4.1.9.9.48.1.2.1.3) Mémoire utilisée depuis 10 min.
- ciscoEnvMonFanState (.1.3.6.1.4.1.9.9.13.1.4.1.3) L'état actuel du ventilateur

Machine serveur ou client :

- hrProcessorLoad (.1.3.6.1.2.1.25.3.3.1.2) La moyenne du pourcentage de temps que le processeur tourne. Permet de calculer la charge CPU
- HrMemorySize (.1.3.6.1.2.1.25.2.2) La taille de la mémoire RAM contenue par l'hôte
- hrSWRunPerfMem (.1.3.6.1.2.1.25.5.1.1.2.x) La mémoire utilisée par processeur
- HrStorageTypes (.1.3.6.1.2.1.25.2.1) Obtenir les identifiants parmi les types de stockage (clé usb, disquette, disque externe, RAM...)
- hrSystemUptime (.1.3.6.1.2.1.25.1.1) : Temps depuis lequel l'hôte est en service
- hrSystemNumUsers (.1.3.6.1.2.1.25.1.5) : Nombre d'utilisateurs sur l'hôte
- HrStorageUsed (.1.3.6.1.2.1.25.2.3.1.6) L'espace disque utilisé
- hrSWRunStatus (.1.3.6.1.2.1.25.4.2.1.7) Affiche l'état d'un service (running, invalid...)

Machine Linux Debian :

CPU Load

- 1 minute Load: .1.3.6.1.4.1.2021.10.1.3.1
- 5 minute Load: .1.3.6.1.4.1.2021.10.1.3.2
- 15 minute Load: .1.3.6.1.4.1.2021.10.1.3.3

CPU

- percentage of user CPU time: .1.3.6.1.4.1.2021.11.9.0
- raw user cpu time: .1.3.6.1.4.1.2021.11.50.0
- percentages of system CPU time: .1.3.6.1.4.1.2021.11.10.0
- raw system cpu time: .1.3.6.1.4.1.2021.11.52.0
- percentages of idle CPU time: .1.3.6.1.4.1.2021.11.11.0
- raw idle cpu time: .1.3.6.1.4.1.2021.11.53.0
- raw nice cpu time: .1.3.6.1.4.1.2021.11.51.0

Memory Statistics

- Total Swap Size: .1.3.6.1.4.1.2021.4.3.0

- Available Swap Space: .1.3.6.1.4.1.2021.4.4.0
- Total RAM in machine: .1.3.6.1.4.1.2021.4.5.0
- Total RAM used: .1.3.6.1.4.1.2021.4.6.0
- Total RAM Free: .1.3.6.1.4.1.2021.4.11.0
- Total RAM Shared: .1.3.6.1.4.1.2021.4.13.0
- Total RAM Buffered: .1.3.6.1.4.1.2021.4.14.0
- Total Cached Memory: .1.3.6.1.4.1.2021.4.15.0

Disk Statistics

- Path where the disk is mounted: .1.3.6.1.4.1.2021.9.1.2.1
- Path of the device for the partition: .1.3.6.1.4.1.2021.9.1.3.1
- Total size of the disk/partion (kBytes): .1.3.6.1.4.1.2021.9.1.6.1
- Available space on the disk: .1.3.6.1.4.1.2021.9.1.7.1
- Used space on the disk: .1.3.6.1.4.1.2021.9.1.8.1
- Percentage of space used on disk: .1.3.6.1.4.1.2021.9.1.9.1
- Percentage of inodes used on disk: .1.3.6.1.4.1.2021.9.1.10.1
- System Uptime: .1.3.6.1.2.1.1.3.0

Pour pouvez également en trouver des centaines d'autres...

7) Commandes SNMP sous Linux

Syntaxe générale :

La syntaxe des commandes SNMP est la suivante :

Exemples :

- ➔ Permet de récupérer le « hrStorageEntry » : affiche tous les espaces de stockage.
- ➔ Permet de récupérer le « hrStorageUsed » : affiche l'utilisation des espaces de stockage.

Options possibles :

- Les résultats s'affichent en format raccourci (avec le nom de la MIB et pas le numéro de l'OID). Pour afficher l'OID complet, il faut préciser l'option -Of (O comme output, sortie ; f comme full, complet).

Exemple :

- Pour afficher les résultats qu'au format numérique : -On (n comme numeric)
- Pour afficher les résultats qu'avec la valeur (sans rappel de l'OID interrogé) : -Ov (v comme value only)
- Pour éviter une traduction de la valeur d'un OID : -Oe
- Pour ne pas afficher le type de la valeur : -OQ

→ Au final, on peut cumuler les options, par exemple : -OevQ

- Pour snmpwalk, on peut avoir l'option -Ci pour afficher l'index du tableau de résultat et -Cb pour afficher les entêtes de colonne sous forme courte (plus lisible). En effet, de nombreux OID sont organisés sous forme de tableau : les partitions, les interfaces réseaux, la liste des processus, etc.

8) Mise en pratique avec Linux Debian

Dans cette mise en pratique, nous allons utiliser une machine Debian qui servira de machine cliente (agent SNMP) sur laquelle on récupèrera des informations, et une autre machine Debian (le serveur web 1) qui sera donc serveur (manager) car c'est elle qui sera interrogée en SNMP et qui répondra aux requêtes. **Pour commencer, on placera la machine cliente sur le même réseau (DMZ interne).**

Nous allons donc installer snmp sur le client et snmpd sur le serveur et les configurer. Puis nous ferons quelques requêtes SNMP manuelles pour voir si tout fonctionne. Ensuite, nous créerons un script pour avoir un petit programme de supervision (qui fera tout seul des requêtes SNMP) en affichant des informations essentielles et lisibles pour un utilisateur standard.

1. Installation du paquet snmp sur la machine Debian « supervision » et du paquet snmpd sur la machine Debian « Serveur Web 1 ».
2. Vérification du service snmpd :

```
root@debian:~# systemctl status snmpd
● snmpd.service - Simple Network Management Protocol (SNMP) Daemon.
   Loaded: loaded (/lib/systemd/system/snmpd.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2019-10-07 09:42:11 CEST; 17min ago
     Process: 8802 ExecStartPre=/bin/mkdir -p /var/run/agentx (code=exited, status=0/SUCCESS)
    Main PID: 8805 (snmpd)
       Tasks: 1 (limit: 4915)
      CGroup: /system.slice/snmpd.service
              └─8805 /usr/sbin/snmpd -Lsd -Lf /dev/null -u Debian-snmp -g Debian-snmp -I -smux mteTrigg

oct. 07 09:42:33 debian snmpd[8805]: Connection from UDP: [127.0.0.1]:58433->[127.0.0.1]:161
oct. 07 09:42:33 debian snmpd[8805]: Connection from UDP: [127.0.0.1]:58433->[127.0.0.1]:161
oct. 07 09:43:01 debian snmpd[8805]: Connection from UDP: [127.0.0.1]:34790->[127.0.0.1]:161
oct. 07 09:43:01 debian snmpd[8805]: Connection from UDP: [127.0.0.1]:34790->[127.0.0.1]:161
oct. 07 09:43:11 debian snmpd[8805]: Connection from UDP: [127.0.0.1]:36127->[127.0.0.1]:161
oct. 07 09:43:11 debian snmpd[8805]: Connection from UDP: [127.0.0.1]:36127->[127.0.0.1]:161
oct. 07 09:56:59 debian snmpd[8805]: Connection from UDP: [127.0.0.1]:47719->[127.0.0.1]:161
oct. 07 09:56:59 debian snmpd[8805]: Connection from UDP: [127.0.0.1]:47719->[127.0.0.1]:161
oct. 07 09:57:03 debian snmpd[8805]: Connection from UDP: [127.0.0.1]:44340->[127.0.0.1]:161
oct. 07 09:57:03 debian snmpd[8805]: Connection from UDP: [127.0.0.1]:44340->[127.0.0.1]:161
lines 1-19/19 (END)
```

3. Modification du fichier de configuration du serveur SNMP :

Décommenter la ligne

afin de lancer l'écoute sur le port 161, en UDP, en provenance de n'importe quelle machine.

Rappeler l'intérêt de communiquer via UDP et pas TCP :

.....
.....
.....

4. Modifier les « view » et les OID associés, et paramétrer la communauté (droits, nom, machines autorisées à contacter la communauté -ici rien après rocommunity damico, mais on aurait pu mettre l'adresse IP de la machine cliente de supervision -) :

```

GNU nano 2.7.4          Fichier : /etc/snmp/snmpd.conf

# then remember to update the other occurrences in this example config file to match.

#####
#
# ACCESS CONTROL
#

view systemview included .1.3.6.1.2.1
view systemview included .1.3.6.1.2.1.25.1

# Full access from the local host
rocommunity damico localhost

# Default access to basic system info rocommunity$
# rocommunity6 is for IPv6 rocommunity6 public $

# Full access from an example network

```

Dire à quoi correspond le droit « ro » avant « community » :

.....

5. Enregistrer, fermer et redémarrer le service :

6. Vérification de l'ouverture du port UDP 161 :

```

root@debian:~# ss -ulnp
State      Recv-Q Send-Q   Local Address:Port      Peer Address:Port
UNCONN     0      0             *:56877                  *:*
users: (('snmpd',pid=8805,fd=10))
UNCONN     0      0             *:68                    *:*
users: (('dhclient',pid=638,fd=6))
UNCONN     0      0      127.0.0.1:161           *:*
users: (('snmpd',pid=8805,fd=9))
root@debian:~#

```

7. Tester une requête SNMPWALK pour vérifier le bon fonctionnement (depuis la machine cliente) :

```

root@debian:~# snmpwalk -v1 127.0.0.1 -c damico .1.3.6.1.2.1.1.0
iso.3.6.1.2.1.1.0 = STRING: "Linux debian 4.9.0-11-amd64 #1 SMP Debian 4.9.189-3 (2019-09-02) x86_64"
root@debian:~# _

```

Une autre requête (avec un autre OID), ici la mémoire RAM restante :

```

root@debian:~# snmpwalk -v1 127.0.0.1 -c damico .1.3.6.1.4.1.2021.4.6
iso.3.6.1.4.1.2021.4.6.0 = INTEGER: 422648
root@debian:~#

```

La même avec cette fois une option pour l'affichage de sortie (on remarquera que c'est plus clair) :

```

root@debian:~# snmpwalk -v1 -Ov 127.0.0.1 -c damico .1.3.6.1.4.1.2021.4.6
INTEGER: 422576
root@debian:~#

```

Idem, avec cette fois juste la réponse désirée (ce qui nous facilitera la vie lors de l'écriture de scripts) :

```
root@debian:~# snmpwalk -v1 -OvQ 127.0.0.1 -c damico .1.3.6.1.4.1.2021.4.6
422396
root@debian:~#
```

Expliquer pourquoi il est intéressant d'utiliser l'option « -OvQ » :

.....

.....

.....

8. Nous allons maintenant créer un script pour exécuter d'un coup plusieurs requêtes SNMP et rendre l'affichage des résultats plus conviviale. Cela ressemblera un peu à un tableau de bord d'un logiciel de supervision...

9. Création d'un dossier pour contenir le script d'automatisation de requêtes SNMP et changement de répertoire pour s'y rendre :

```
root@debian:~# mkdir /scripts
root@debian:~# cd /scripts/
root@debian:/scripts# _
```

10. Création du fichier « supervision » qui contiendra le script et changement des droits pour le rendre exécutable :

```
root@debian:/scripts# touch supervision
root@debian:/scripts# chmod u+x supervision
root@debian:/scripts#
```

Remarque : si on précise l'extension .sh, l'éditeur pour afficher des couleurs, ce qui facilitera la lecture du code.

11. Création d'une ligne d'affichage dans le script et exécution de celui-ci :

Ouvrir le fichier dans l'éditeur puis saisir : echo "Programme de supervision de la machine Debian ...".

Enregistrer et exécuter le script (bash nomDuScript ou ./nomDuScript)

```
root@debian:/scripts# bash supervision
Programme de supervision de la machine Debian DAMICO
root@debian:/scripts# _
```

12. Ecriture du script :

Requêtes SNMPWALK pour récupérer la RAM total de la machine et la RAM disponible.

Affichage en méga-octets et message d'alerte en fonction du résultat.

```

GNU nano 2.7.4                                Fichier : supervision
echo -e "\n"
echo "Programme de supervision de la machine Debian DAMICO"
echo -e "\n"

echo "*****MEMOIRE*****"
memoireTotale=`snmpwalk -v1 127.0.0.1 -OvQ -c damico .1.3.6.1.4.1.2021.4.5`
echo "RAM totale : " $memoireTotale

memoireLibre=`snmpwalk -v1 127.0.0.1 -OvQ -c damico .1.3.6.1.4.1.2021.4.6`
echo "RAM disponible :" $memoireLibre
echo "*****"

memoireLibreMega=$(( $memoireLibre/1024 ))
echo "Il reste " $memoireLibreMega "méga-octets de RAM"
if [ $memoireLibreMega -lt 500 ]
then
    echo "ALERTE RAM !"
fi

echo -e "\n\n"

```

Expliquer le contenu du script :

.....

.....

.....

.....

13. Affichage du script :

```

root@debian:/scripts# ./supervision

Programme de supervision de la machine Debian DAMICO

*****MEMOIRE*****
RAM totale : 1020384
RAM disponible : 418656
*****
Il reste 408 méga-octets de RAM
ALERTE RAM !

```

14. Compléter le script pour afficher des informations sur le CPU et le disque dur, avec un rafraîchissement automatique des informations toutes les 10 secondes.