

Déploiement de Nagios et RANCID : Supervision et Sauvegarde des Actifs Réseaux

Société menuimetal

Présenté par :

BAYERE Abdoul Fatahou
GUIHARD Mathieu

Sommaire


I. Introduction.....	1
II. Mise en place du Gantt.....	1
III. Mise à jour du schéma réseau.....	2
IV. Mission 0 : Prérequis et contraintes.....	2
A. Créer ou cloner deux VMs serveurs linux.....	2
B. Référencement des machines dans le GLPI et DNS.....	3
1. Référencement dans le dns.....	3
2. Référencement dans glpi.....	3
3. Test de la résolution des nom du DNS.....	4
C. Test de connexion en SSH.....	5
D. Installation et configuration de Nagios Core.....	5
V. Mission 1 : NAGIOS.....	14
A. Étude sur l'utilisation de Nagios comme solution de supervision.....	14
B. Supervision sur un client windows.....	17
C. Installation de Check NCPA sur le serveur Nagios.....	23
1. Installation du plugin check_ncpa sur le serveur Nagios.....	24
2. Test des commandes check_ncpa.....	26
3. Ajout de la config sur nagios.....	27
D. Configuration du lien du client windows dans le serveur.....	28
E. Modification de l'utilisateur admin.....	34
F. Supervision sur un client linux.....	35
1. Installation et configuration de NRPE (Serveur nagios).....	35
G. Étapes pour installer et configurer NRPE sur l'hôte client Debian.....	41
H. Étapes pour Surveiller les hôtes Linux sur le serveur.....	42
1. Configuration des templates.....	43
I. Configuration de l'hôte dans client-debian.cfg.....	44
J. Intégration et configuraiton des templates services.....	46
1. Installation des plugins nécessaire pour la supervision.....	49
K. Intégration du Service GLPI dans Nagios.....	49
VI. MISSION 2 : Sauvegarde d'actifs réseaux avec Rancid.....	52
A. Installation de TFTP.....	54
B. Configuration du switch HP et récupération de la config via TFTP.....	55

Liens vers les ressources partagées :

Gestion VMs et VLANS :

 (GUIHARD_BAYERE) Gestion VLAN et VMs du contexte "Menui..."

Gantt :  (GUIHARD_BAYERE) Diagramme de Gantt AP6

 Suite à un impératif familial, Mathieu, qui gère la partie Rancid et Switchs, a dû s'absenter durant les 3 dernières heures du TP.

I. Introduction

En 1980, Jean Morin crée Menuimetal.SA à Lens, une entreprise spécialisée dans la conception et la fabrication de structures en métal et en verre. Tout en se concentrant sur la production de huisseries et d'éléments de façade, Menuimetal délègue la pose à des partenaires externes. Avec un bureau d'étude capable de répondre aux besoins spécifiques de ses clients, l'entreprise propose des solutions sur mesure et poursuit sa croissance en cherchant à optimiser ses services informatiques.

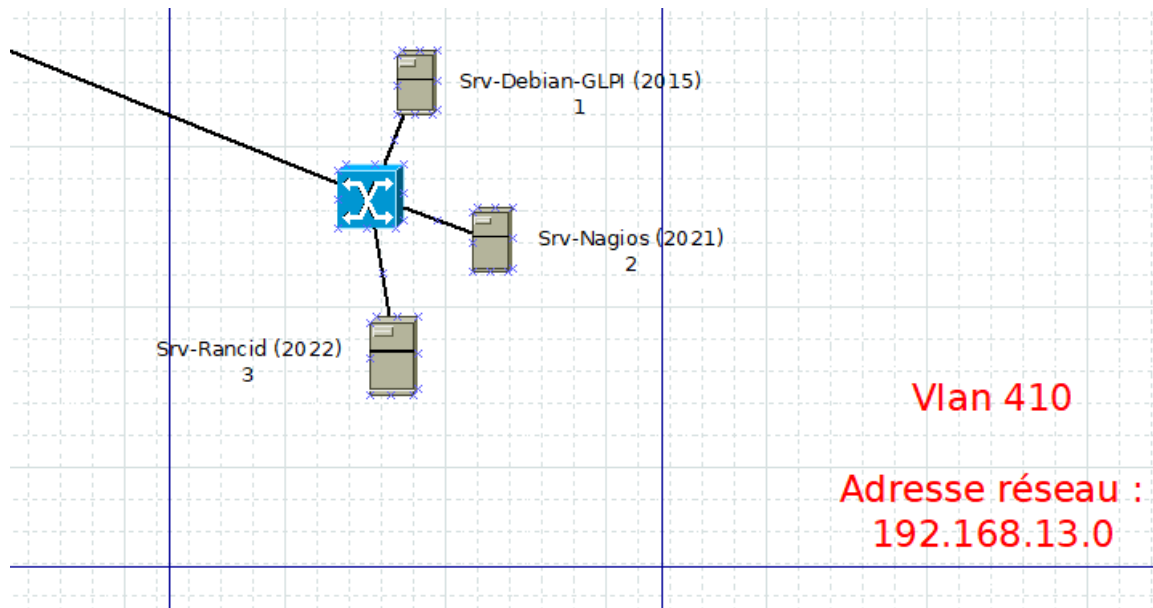
En tant que stagiaire chez MenuiMétal, sous la direction d'Olivier Lepage et avec l'accompagnement de votre tutrice Catherine Bercy, nous allons travailler à la mise en place d'un service de supervision et à l'intégration de RANCID (Really Awesome New Cisco confg Differ). Ce projet permettra d'améliorer la surveillance des infrastructures informatiques de l'entreprise et d'automatiser la gestion des configurations réseau, contribuant ainsi à l'efficacité et à la sécurité des systèmes.

II. Mise en place du Gantt

Tâches ou WBS					
Lettre	Titre	Jour et heure de début	Antécédent(s)	Durée en heure	Affectée à
A	Créer/Cloner 2 VMs serveurs Linux	08/10/2024 14:00:00		0.5	Mathieu
B	Référencer VMs sur DNS et GLPI	08/10/2024 14:30:00	A	0.5	Abdoul et Mathieu
C	Mettre à jour schéma réseau	08/10/2024 15:00:00	B	0.5	Abdoul et Mathieu
D	Configuration et Accès SSH aux VMs	08/10/2024 15:30:00	C	0.5	Abdoul et Mathieu
E	Étude sur Nagios	08/10/2024 16:00:00	F	0.5	Abdoul
F	Installer Nagios et configuration	08/10/2024 16:00:00	E	2	Abdoul
G	Installer RANCID	08/10/2024 15:30:00	F	0.5	Mathieu
H	Installer TFTPd hpa	08/10/2024 16:00:00	G	0.5	Mathieu
I	Configurer la supervision avec Nagios	10/10/2024 08:30:00	F,H	4	Abdoul
J	Configurer switch HP	10/10/2024 08:30:00	I	4	Abdoul et Mathieu
K	Configurer notifications mail et Nagios	10/10/2024 13:30:00	J	4	Abdoul
L	Configurer switch Cisco	10/10/2024 13:30:00	K	4	Abdoul et Mathieu
M	Backup manuel des switches	11/10/2024 13:30:00	L	2	Abdoul et Mathieu
N	Tester la restauration	11/10/2024 15:30:00	M	1	Abdoul et Mathieu
O					A la fin
					Pens
					Pour
					Dans

Visualisation du Gantt

III. Mise à jour du schéma réseau



Visualisation du schéma réseau mis à jour

IV. Mission 0 : Prérequis et contraintes

A. Créer ou cloner deux VMs serveurs linux,

Créer ou cloner deux VMs serveurs linux, modifier leur hostname, changer leur adresse IP. Les VMs sont dans le vlan GESTION (410) :

- 2021 (Srv-Nagios)
- 2022 (Srv-Rancid)

Network Device (net0) virtio=BC:24:11:0E:95:8E,bridge=vibr0,tag=410

Network Device (net0) virtio=BC:24:11:36:96:C2,bridge=vibr0,tag=410

```
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether bc:24:11:0e:95:8e brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 192.168.13.2/24 brd 192.168.13.255 scope global ens18
        valid_lft forever preferred_lft forever
    inet6 fe80::be24:11ff:fe0e:950e/64 scope link
        valid_lft forever preferred_lft forever
root@srv-nagios:~#
```

```

2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether bc:24:11:36:96:c2 brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 192.168.13.3/24 brd 192.168.13.255 scope global ens18
        valid_lft forever preferred_lft forever
    inet6 fe80::be24:11ff:fe36:96c2/64 scope link
        valid_lft forever preferred_lft forever
root@srv-rancid:~#

```

Modification du nom de machine

B. Référencement des machines dans le GLPI et DNS

1. Référencement dans le dns

```

srv-nagios IN A 192.168.13.2
srv-rancid IN A 192.168.13.3
~

```

Référencement des machines dans le DNS

2. Référencement dans glpi

Pour le référencement dans le GLPI on procédera ainsi :

- On applique le proxy dans le fichier **/etc/wgetrc**

```

# You can set the default proxies for Wget to use for http, https, and ftp.
# They will override the value in the environment.
https_proxy = http://172.16.0.35:3142/
http_proxy = http://172.16.0.35:3142/
#ftp_proxy = http://proxy.yoyodyne.com:18023/

```

- On exécute la commande wget pour télécharger l'agent :
- On applique le proxy dans le fichier **/etc/wgetrc**
- On exécute la commande wget pour télécharger l'agent :

wget

<https://github.com/glpi-project/glpi-agent/releases/download/1.7.1/glpi-agent-1.7.1-linux-installer.pl>

- On installe Perl qui est un langage de scripting :

apt install perl

Puis on installe l'agent glpi :

perl [glpi-agent-1.7.1-linux-installer.pl](#) -s <http://192.168.13.1/> --runnow --install

Enfin on vérifie que l'agent est bien actif sur le client et que ce dernier est bien détecté sur le serveur :

systemctl status glpi-agent

<input type="checkbox"/> srv-nagios	QEMU	QEMU	Standard PC (i440FX + PIIX, 1996)	Debian GNU/Linux 12 (bookworm)	2024-10-08 12:51	QEMU Virtual CPU version 2.5+
<input type="checkbox"/> srv-rancid	QEMU	QEMU	Standard PC (i440FX + PIIX, 1996)	Debian GNU/Linux 12 (bookworm)	2024-10-08 12:55	QEMU Virtual CPU version 2.5+

3. Test de la résolution des nom du DNS

```
root@dns:~# nslookup srv-nagios
Server:      192.168.12.1
Address:     192.168.12.1#53

Name:   srv-nagios.menuimetal.fr
Address: 192.168.13.2

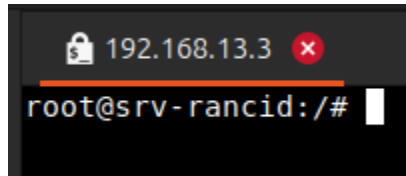
root@dns:~# nslookup srv-rancid
Server:      192.168.12.1
Address:     192.168.12.1#53

Name:   srv-rancid.menuimetal.fr
Address: 192.168.13.3

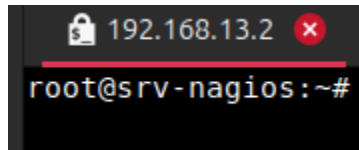
root@dns:~# _
```

Test avec nslookup

C. Test de connexion en SSH



Test de connexion sur Rancid



Test de connexion sur Nagios

D. Installation et configuration de Nagios Core

```
apt update
```

```
root@srv-nagios:~# apt install wget unzip curl openssl build-essential libgd-dev libssl-dev libapache2-mod-php php-gd php apache2 -y
```

Téléchargement les fichiers d'installation de Nagios Core

Remarque : Il est à noter que le **daemon** s'installe simultanément avec les autres services inclus dans la commande suivante :

```
apt install -y build-essential apache2 php openssl perl  
make php-gd libgd-dev libapache2-mod-php libperl-dev  
libssl-dev daemon wget apache2-utils unzip.
```


Pour télécharger la dernière version, nous avons utilisé le site [officiel des versions](#).

Lien utilisé :

<https://github.com/NagiosEnterprises/nagioscore/releases/download/nagios-4.5.5/nagios-4.5.5.tar.gz>

```
root@srv-nagios:~# wget https://github.com/NagiosEnterprises/nagioscore/releases/download/nagios-4.5.5/nagios-4.5.5.tar.gz
```

Version de nagios 4.5.5

```
root@srv-nagios:~# tar -zxvf nagios-4.5.5.tar.gz
```

Décompression de Nagios

```
root@srv-nagios:~# cd nagios-4.5.5  
root@srv-nagios:~/nagios-4.5.5#
```

Déplacement dans le dossier Nagios

```
root@srv-nagios:~/nagios-4.5.5# ./configure
```

Exécution du script de configuration de Nagios Core

```

config.status: creating lib/snprintf.h
config.status: creating lib/iobroker.h

Creating sample config files in sample-config/ ...

*** Configuration summary for nagios 4.5.5 2024-09-17 ***:

General Options:
-----
Nagios executable: nagios
Nagios user/group: nagios,nagios
Command user/group: nagios,nagios
Event Broker: yes
Install ${prefix}: /usr/local/nagios
Install ${includedir}: /usr/local/nagios/include/nagios
Lock file: /run/nagios.lock
Check result directory: /usr/local/nagios/var/spool/checkresults
Init directory: /lib/systemd/system
Apache conf.d directory: /etc/apache2/sites-available
Mail program: /bin/mail
Host OS: linux-gnu
IOBroker Method: epoll

Web Interface Options:
-----
HTML URL: http://localhost/nagios/
CGI URL: http://localhost/nagios/cgi-bin/
Traceroute (used by WAP):

Review the options above for accuracy. If they look okay,
type 'make all' to compile the main program and CGIs.

root@srv-nagios:~/nagios-4.5.5#

```

Nous avons configuré Nagios 4.5.5 sur le serveur et voici le récapitulatif de la configuration :

Nagios est installé dans **/usr/local/nagios**, un répertoire dédié aux logiciels compilés manuellement pour éviter les conflits avec ceux installés via le gestionnaire de paquets Debian.

L'exécutable principal est nagios, et il s'exécute sous **l'utilisateur et le groupe nagios**, pour limiter les permissions et sécuriser l'application. Les commandes, comme les vérifications d'état des services et des hôtes, sont également lancées par cet utilisateur.

Chemins importants :

- Les fichiers de configuration seront situés dans **/usr/local/nagios/etc**.
- Les résultats des vérifications seront stockés dans **/usr/local/nagios/var/spool/checkresults**.

Méthode de gestion des événements :

- Nagios utilisera epoll pour gérer les événements d'entrée/sortie, ce qui est optimisé pour les systèmes Linux. Cela va être particulièrement utile pour gérer efficacement les nombreuses connexions réseau à surveiller.

Interface web :

- L'interface web sera disponible à l'adresse **http://localhost/nagios/**.

Maintenant que la configuration est en place, je vais pouvoir lancer make all pour compiler Nagios et finaliser l'installation avec make install.

Compilation du programme principal et des CGI :

```
root@srv-nagios:~/nagios-4.5.5# make all
cd ./base && make
make[1] : on entre dans le répertoire « /root/nagios-4.5.5/base »
gcc -Wall -I.. -I. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o na
gios.o ./nagios.c
gcc -Wall -I.. -I. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o br
oker.o broker.c
gcc -Wall -I.. -I. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o ne
bmods.o nebmods.c
gcc -Wall -I.. -I. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o ..
/common/shared.o ../common/shared.c
gcc -Wall -I.. -I. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o qu
ery-handler.o query-handler.c
gcc -Wall -I.. -I. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o wo
rkers.o workers.c
gcc -Wall -I.. -I. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o ch
ecks.o checks.c
gcc -Wall -I.. -I. -I../lib -I../include -I../include -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o co
nfig.o config.c
```

Création d'un groupe et un utilisateur :

```
root@srv-nagios:~/nagios-4.5.5# make install-groups-users
groupadd -r nagios
useradd -g nagios nagios
root@srv-nagios:~/nagios-4.5.5#
```

```
sudo usermod -aG www-data nagios
```

On lance l'installation avec les commandes suivantes :

```
sudo make install
sudo make install-init
sudo make install-commandmode
sudo make install-config
sudo make install-webconf
sudo a2enmod rewrite
sudo a2enmod cgi
```

On redémarre le service Apache :

```
sudo systemctl restart apache2
```

```

root@srv-nagios:~/nagios-4.5.5# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Wed 2024-10-09 11:20:07 CEST; 18s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 587 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
    Main PID: 591 (apache2)
      Tasks: 6 (limit: 2306)
         Memory: 14.0M
            CPU: 94ms
       CGroup: /system.slice/apache2.service
               └─591 /usr/sbin/apache2 -k start
                 └─592 /usr/sbin/apache2 -k start
                   └─593 /usr/sbin/apache2 -k start
                     └─594 /usr/sbin/apache2 -k start
                       └─595 /usr/sbin/apache2 -k start
                         └─596 /usr/sbin/apache2 -k start

oct. 09 11:20:07 srv-nagios systemd[1]: Starting apache2.service - The Apache HTTP Server.
oct. 09 11:20:07 srv-nagios apachectl[590]: AH00557: apache2: apr_sockaddr_info_get() fail
oct. 09 11:20:07 srv-nagios apachectl[590]: AH00558: apache2: Could not reliably determine
oct. 09 11:20:07 srv-nagios systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-21/21 (END)

```

Création du compte Administrateur :

login : admin

mdp : admin

```

root@srv-nagios:~/nagios-4.5.5# htpasswd -c /usr/local/nagios/etc/htpasswd.users admin

```

```

root@srv-nagios:~/nagios-4.5.5# htpasswd -c /usr/local/nagios/etc/htpasswd.users admin
New password:
Re-type new password:
Adding password for user admin
root@srv-nagios:~/nagios-4.5.5#

```

Edition fichiers contact :

Afin de recevoir des alertes, notre fichier `contacts.cfg`, situé à `/usr/local/nagios/etc/objects/contacts.cfg`, doit être édité pour remplacer `nagios@localhost` par l'adresse e-mail de notre choix.

On vérifie que l'utilisateur **nagiosadmin** est déjà configuré :

```
define contact |
    contact_name      nagiosadmin          ; Short name of user
    use               generic-contact      ; Inherit default values from generic-contact template (defined above)
    alias             Nagios Admin         ; Full name of user
    email             nagios@localhost ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS *****
```

Activation de la surveillance des serveurs distants dans /usr/local/nagios/etc/nagios.cfg :

Supprimez le commentaire de la ligne dans /usr/local/nagios/etc/nagios.cfg pour permettre la surveillance des serveurs distants.

cfg_dir=/usr/local/nagios/etc/servers

```
cfg_dir=/usr/local/nagios/etc/servers
#cfg_dir=/usr/local/nagios/etc/printers
#cfg_dir=/usr/local/nagios/etc/switches
```

Puis on va créer un répertoire serveur :

sudo mkdir -p /usr/local/nagios/etc/servers

```
root@srv-nagios:~# mkdir -p /usr/local/nagios/etc/servers
root@srv-nagios:~#
```

Activation des modules Apache pour le serveur Nagios :

```
root@srv-nagios:~# a2enmod rewrite
Module rewrite already enabled
root@srv-nagios:~# a2enmod cgi
Module cgi already enabled
root@srv-nagios:~# |
```

Redémarrage du serveur Apache et lancement de Nagios Core :

```
root@srv-nagios:~# service apache2 restart
root@srv-nagios:~# service nagios restart
root@srv-nagios:~#
```

Vérification de la configuration :

On vérifie la syntaxe de la configuration en exécutant la commande suivante :

```
root@srv-nagios:~/nagios-4.5.5# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

```
root@srv-nagios:~/nagios-4.5.5# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.5.5
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-09-17
License: GPL

Website: https://www.nagios.org
Reading configuration data...
  Read main config file okay...
  Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
  Checked 8 services.
  Checked 1 hosts.
  Checked 1 host groups.
  Checked 0 service groups.
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 24 commands.
  Checked 5 time periods.
  Checked 0 host escalations.
  Checked 0 service escalations.
Checking for circular paths...
  Checked 1 hosts
  Checked 0 service dependencies
  Checked 0 host dependencies
  Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
root@srv-nagios:~/nagios-4.5.5#
```

Activation du service Nagios au démarrage du système :

```
root@srv-nagios:~/nagios-4.5.5# systemctl enable nagios
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /lib/systemd/system/nagios.service.
root@srv-nagios:~/nagios-4.5.5#
```

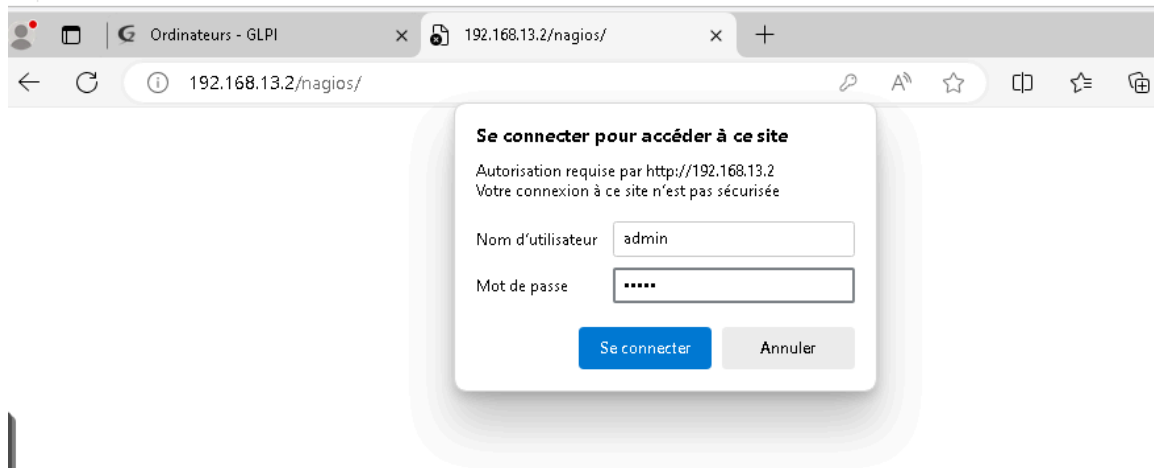
Démarrage du service Nagios :

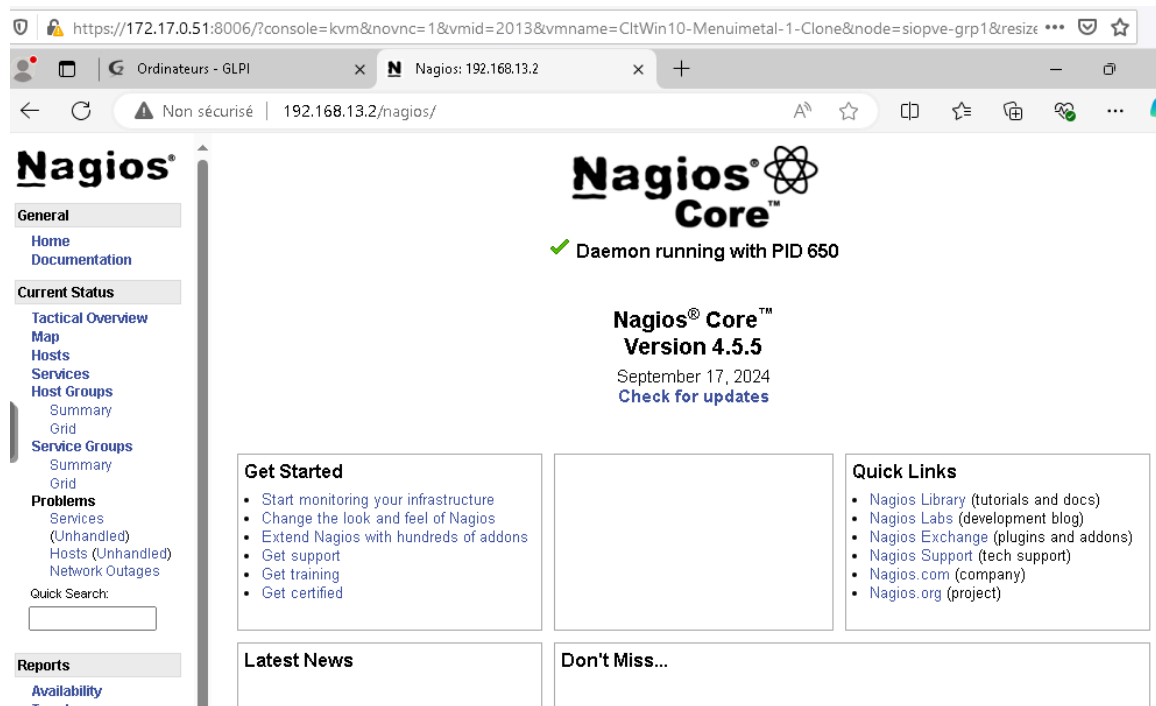
```
s.service.
root@srv-nagios:~/nagios-4.5.5# systemctl start nagios
root@srv-nagios:~/nagios-4.5.5#
```

Accès à l'interface Web de Nagios via le navigateur :

Pour finir, ouvrez un navigateur Web et accédez à l'interface Web de Nagios via l'URL :

http://ServerIP/nagios. Par exemple :





Installation des plugins Nagios :

```
root@srv-nagios:~# apt install nagios-plugins -y
```

V. Mission 1 : NAGIOS

A. Étude sur l'utilisation de Nagios comme solution de supervision

La supervision est un enjeu majeur pour garantir la disponibilité ainsi que le bon fonctionnement d'un Système d'Information (SI), en particulier lorsqu'il s'agit d'une infrastructure comprenant divers éléments tels que des serveurs web, des services externes ou des systèmes intranet. Dans ce contexte, **Nagios** se présente comme l'une des solutions les plus adaptées et performantes du marché pour répondre à ces exigences.

Dans un environnement complexe comme le vôtre, qui comprend plusieurs services critiques (web, intranet, bases de données), il est primordial d'avoir une vue d'ensemble en temps réel sur le fonctionnement de chaque composant. La supervision vous permet :

- **D'identifier les pannes** avant qu'elles n'affectent les utilisateurs finaux.
- **De réagir rapidement** grâce à des alertes en temps réel.
- **D'optimiser les performances** en surveillant les ressources (CPU, mémoire, bande passante, etc.).

Qu'est-ce que Nagios et pourquoi le choisir ?

Nagios est une solution open-source qui est utilisée dans de nombreuses entreprises et projets pour superviser les systèmes informatiques. Il est largement adopté grâce à sa robustesse et sa flexibilité. Voici pourquoi il est particulièrement adapté pour votre SI :

- **Surveillance des services critiques** : Nagios peut surveiller l'état de vos serveurs web, bases de données, services d'intranet, ainsi que des services externes que vous utilisez, en vous permettant de prévenir les défaillances.
- **Alertes en temps réel** : Dès qu'une anomalie est détectée (ex. un serveur web tombe en panne), Nagios envoie immédiatement des notifications (email, SMS) aux administrateurs pour une prise de décision rapide.
- **Large compatibilité** : Nagios peut surveiller un large éventail de systèmes d'exploitation et d'applications, ce qui le rend adapté à votre infrastructure hétérogène.

Tableau comparatif entre Nagios Core (gratuit) et Nagios XI (payant) :

Caractéristique	Nagios Core (Gratuit)	Nagios XI (Payant)
Prix	Gratuit (open-source)	Payant (licence commerciale)
Installation	Manuelle via ligne de commande	Simplifiée avec des assistants
Interface utilisateur	Basique et rudimentaire	Moderne et intuitive
Configuration	Fichiers texte	Interface graphique conviviale
Rapports	Rapports de base	Rapports avancés (SLA, tendances)
Alertes	Notifications basiques	Notifications avancées et personnalisées
Support	Communauté	Support technique officiel
Évolutivité	Très personnalisable (plug-ins)	Préconfiguré avec options avancées

Pour répondre aux besoins de la société **Menuimetal**, nous utiliserons **Nagios Core** (version gratuite). Cette version est parfaitement adaptée à votre infrastructure, qui nécessite une solution robuste pour surveiller vos serveurs web, services intranet et externes sans coûts supplémentaires.

Voici pourquoi **Nagios Core** convient à Menuimetal :

- **Coût réduit** : En tant qu'outil open-source, Nagios Core est entièrement gratuit, ce qui permet de répondre à vos besoins sans investir dans une solution commerciale.

- **Flexibilité** : Il offre une grande extensibilité grâce à sa large bibliothèque de plug-ins, permettant de surveiller vos serveurs, bases de données, et autres services critiques.
- **Configuration personnalisable** : Bien que la configuration se fasse via des fichiers texte, elle permet une personnalisation fine des éléments supervisés, ajustée aux besoins spécifiques de Menuimetal.
- **Notifications en temps réel** : Comme Nagios XI, Nagios Core permet d'envoyer des alertes en cas de problème, garantissant une réactivité immédiate.

En résumé, **Nagios Core** offre toutes les fonctionnalités nécessaires à une supervision efficace du système d'information sans coût additionnel, ce qui en fait le choix idéal pour Menuimetal.

B. Supervision sur un client windows

Installation de NCPA :

NCPA Features Downloads Documentation FAQ

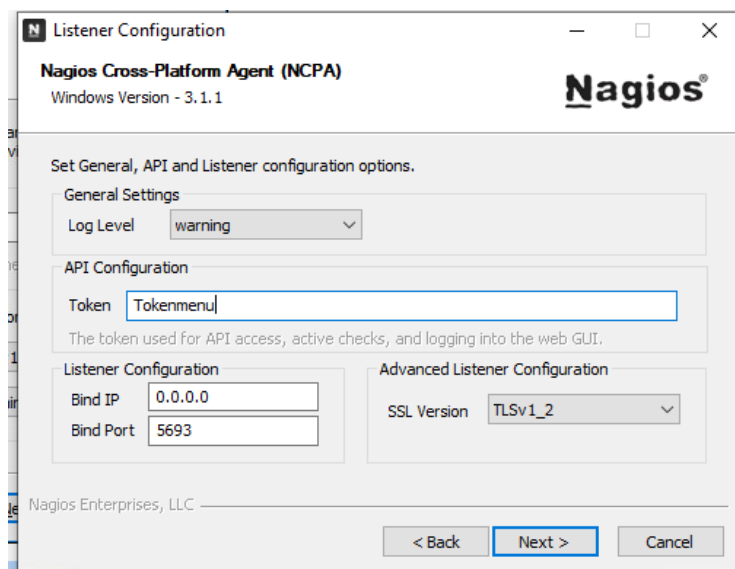
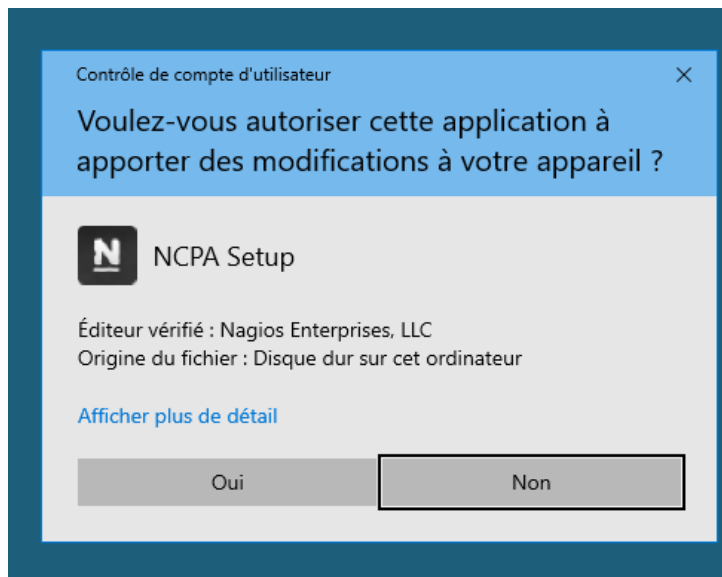
Téléchargements
nca-latest.exe
Ouvrir un fichier

Downloads

Latest stable agent version - **3.1.1** - View the [changelog](#) to see a list all features and bug fixes.
Don't see your version of OS on this list? Visit our [GitHub](#) to build it for your own distribution!

NOTE: before upgrading to NCPA 3 from NCPA 2 on Linux, you must upgrade your GPG key as can be seen at [repo.nagios.com](#).

Windows	RPM Linux	DEB Linux	MacOS
EXE Installer - 64-bit	Install Using Nagios Repo - Recommended	Install Using Nagios Repo - Recommended	DMG Installer - 64-bit
Windows 10+ Windows Server 2016 +	RPM EL 8+ SuSE 15+	DEB - 64bit Debian 11+ Ubuntu 20+	13+



Remarque pour le token :

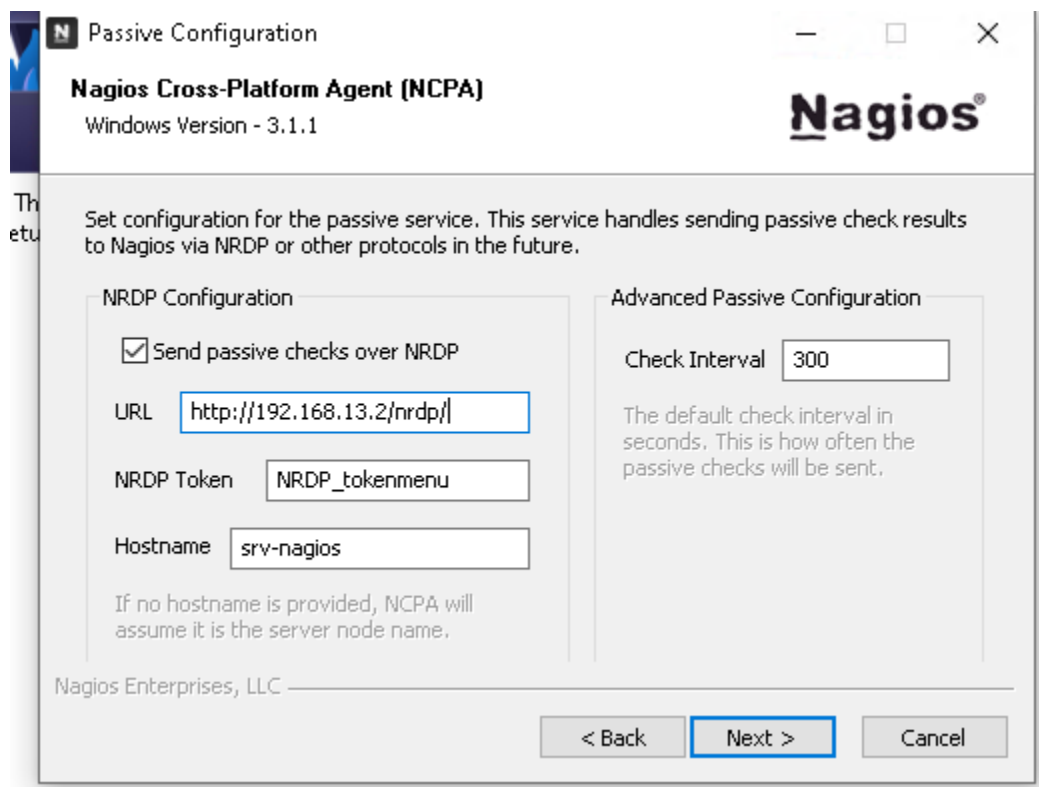
Lors de la configuration des vérifications passives avec NCPA sur une machine, un **jeton d'authentification** est requis pour sécuriser la communication entre le serveur Nagios et l'agent NCPA. Ce jeton agit comme une clé secrète qui garantit que seules les requêtes provenant du serveur Nagios autorisé seront acceptées. On peut définir ce jeton

librement, mais il est recommandé d'utiliser une chaîne complexe pour renforcer la sécurité.

Voici un exemple de jeton sécurisé basé sur le mot menuimetal :

menuimetal_897\$Xz!menui

Dans notre cas, nous utiliserons un nom de jeton plus simple à saisir afin de gagner du temps.



Passive Configuration

Nagios Cross-Platform Agent (NCPA)
Windows Version - 3.1.1

Set configuration for the passive service. This service handles sending passive check results to Nagios via NRDP or other protocols in the future.

NRDP Configuration

☒ Send passive checks over NRDP

URL

NRDP Token

Hostname

If no hostname is provided, NCPA will assume it is the server node name.

Advanced Passive Configuration

Check Interval

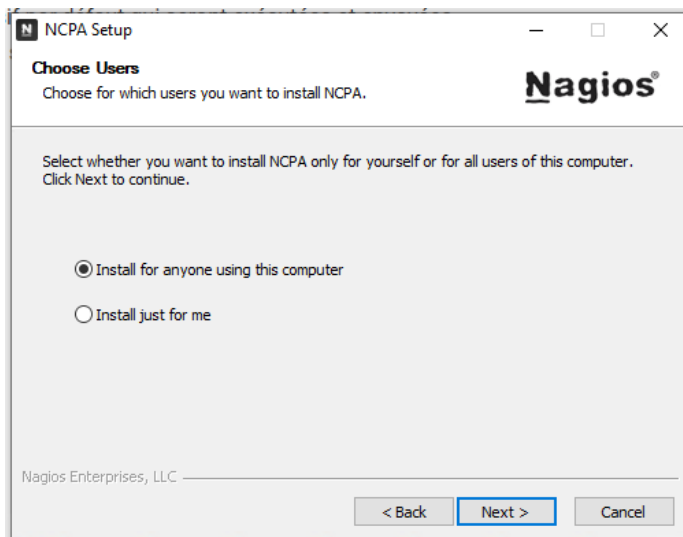
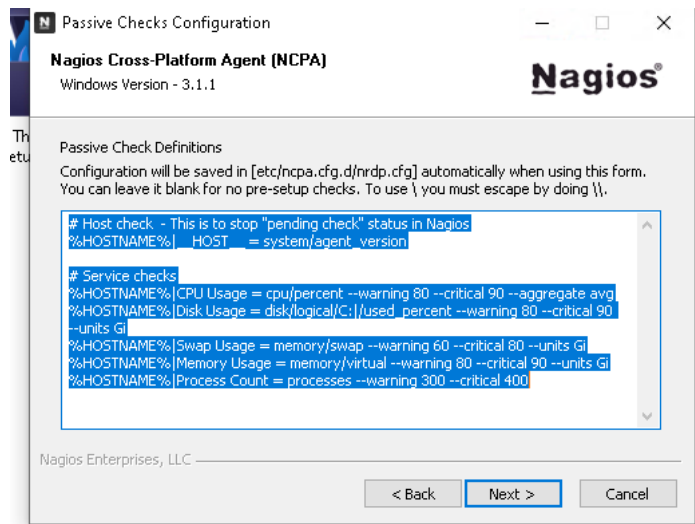
The default check interval in seconds. This is how often the passive checks will be sent.

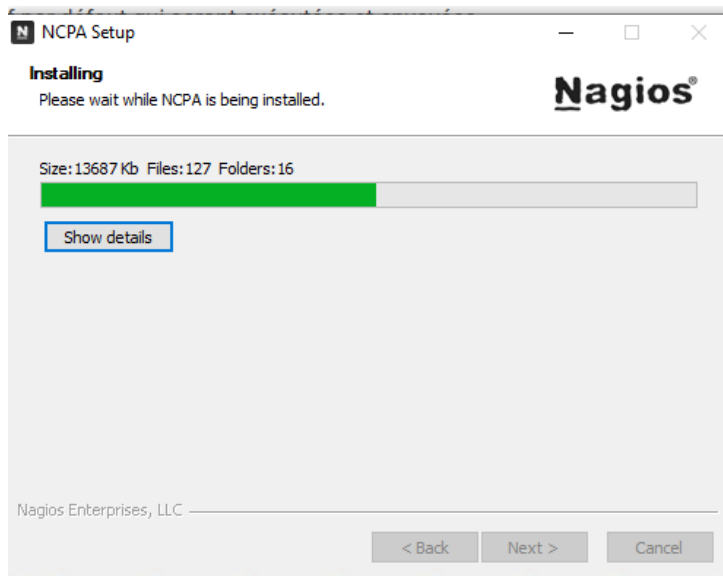
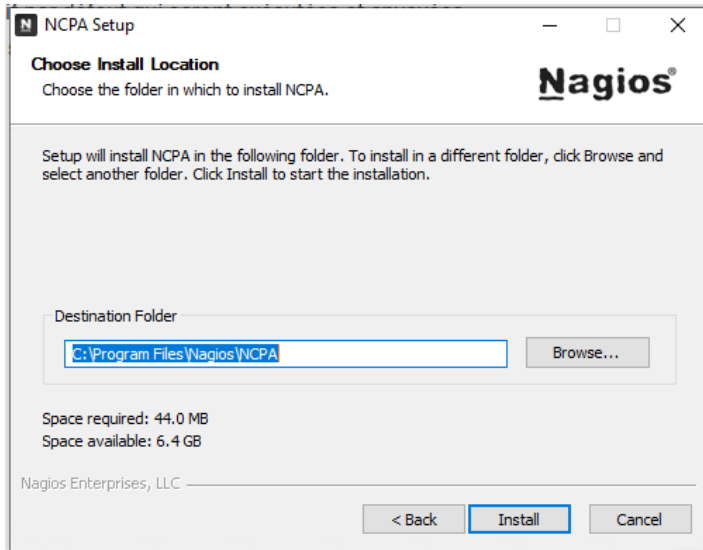
Nagios Enterprises, LLC

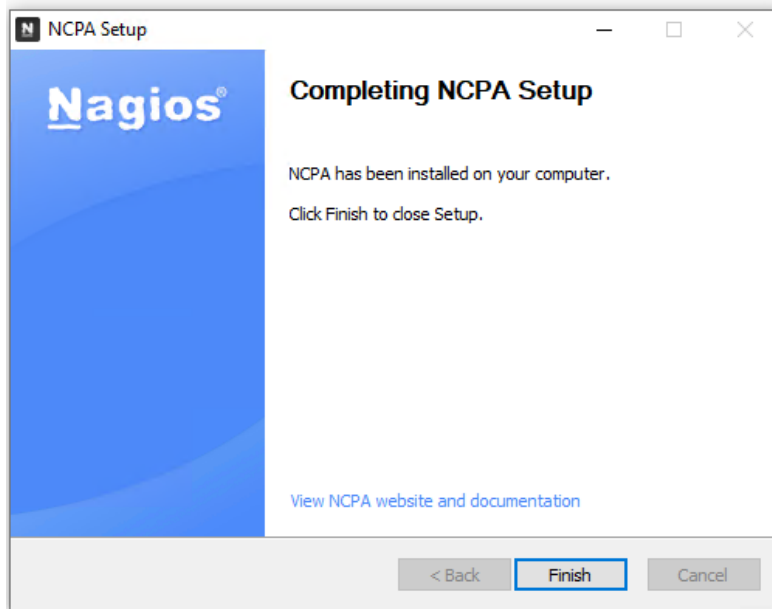
< Back Next > Cancel

Dans l'écran de configuration des vérifications passives, il est essentiel d'activer l'envoi des contrôles passifs via NRDP en cochant la case correspondante. Cela permet à Nagios de recevoir les résultats des vérifications passives exécutées sur des machines distantes. Pour que ces contrôles soient acceptés et traités par Nagios, plusieurs paramètres NRDP (Nagios Remote Data Processor) doivent être fournis :

- L'**URL** fait référence à l'adresse de notre serveur Nagios, là où les résultats des contrôles passifs seront envoyés.
- Le **Jeton NRDP** est une clé d'authentification utilisée pour sécuriser l'envoi des résultats passifs vers Nagios.
- Le **Nom d'hôte** est le nom de la machine sur laquelle les vérifications passives s'exécutent.







Connexion sur l'interface web de NCPA :



Le mot de passe correspond au jeton : Tokenmenu

The screenshot shows the NCPA web interface. The top navigation bar includes links for Dashboard, Checks, Live Data, API, Graphs, Admin, Help, and Logout. The main content area is divided into two sections:

Check Statistics

Check Results	6	(Last 30 days)
---------------	---	----------------

[See Live Stats >](#)

NCPA Agent Information

NCPA Version	3.1.1
Bundled Python	3.12.6 (tags/v3.12.6:a4a2d2b, Sep 6 2024, 20:11:23) [MSC v.1940 64-bit (AMD64)]
Bundled OpenSSL	OpenSSL 3.0.15 3 Sep 2024
Bundled zlib	1.3.1
Agent Uptime	0:01:30

System Information

Node Name	client1
System	Windows 10 10.0.19045
Processor Type	Intel64 Family 15 Model 107 Stepping 1, GenuineIntel

C. Installation de Check NCPA sur le serveur Nagios

Une fois le lien établi côté client, nous allons installer et configurer le plugin **check_ncpa** afin de réaliser des tests.

L'installation du plugin **check_ncpa** sur notre serveur Nagios est essentielle pour permettre à Nagios d'effectuer des vérifications actives sur les systèmes distants où l'agent NCPA est déployé. **NCPA** collecte des informations telles que l'utilisation du CPU, de la mémoire, et du disque sur les machines distantes. Pour que Nagios récupère ces données, il interroge l'agent NCPA via le plugin **check_ncpa**.

Sans ce plugin, Nagios ne peut pas communiquer avec NCPA ni effectuer les vérifications actives nécessaires à la surveillance.

1. Installation du plugin check_ncpa sur le serveur Nagios

Pour installer le plugin, allez sur :

<https://www.nagios.org/ncpa/#downloads>

```
wget https://assets.nagios.com/downloads/ncpa/check_ncpa.tar.gz
```

Extraction :

```
root@srv-nagios:~# tar -xvzf check_ncpa.tar.gz
check_ncpa.py
CHANGES.rst
```

On le déplace vers le répertoire qui contient des plugins :

```
root@srv-nagios:~# mv check_ncpa.py /usr/lib/nagios/plugins/
root@srv-nagios:~#
```

Ensuite, déplacez-le dans le répertoire libexec :

```
root@srv-nagios:~# mv check_ncpa.py /usr/local/nagios/libexec/
```

Remarque : j'aurais pu simplement copier le fichier une fois qu'il avait été placé dans le répertoire

Vérification de la présence de check_ncpa dans le répertoire correct :

```
root@srv-nagios:~# ls /usr/local/nagios/libexec/ | grep "check_ncpa"
check_ncpa.py
root@srv-nagios:~#
```

Ajout des droits d'exécution au plugin installé :

```
root@srv-nagios:/usr/lib/nagios/plugins# chmod a+x check_ncpa.py
root@srv-nagios:/usr/lib/nagios/plugins#
```

Cette manipulation a également été effectuée pour les fichiers contenus dans le répertoire libexec.

Remarque : Dans les images ci-dessus, j'ai placé **le fichier check_ncpa** à la fois dans les dossiers plugins et libexec pour plusieurs raisons. La première est qu'en cours, nous avons appris que les fichiers du dossier plugins devaient être placés dans libexec. La deuxième est qu'en suivant un tutoriel sur YouTube, la personne avait configuré son serveur nagios en laissant les fichiers dans le dossier plugins. Par conséquent, dans le doute, j'ai appliqué à la fois ce que nous avons vu en cours et ce qui était présenté dans la vidéo.

Installation de Python 3 si non présent :

Si Python 3 n'est pas installé sur notre système, on devra l'installer avant de pouvoir utiliser le plugin **check_ncpa**. Pour installer Python 3, on utilisera la commande : `apt install python 3`

Changement du propriétaire du fichier :

```
root@srv-nagios:/usr/lib/nagios/plugins# chown nagios check_ncpa.py
root@srv-nagios:/usr/lib/nagios/plugins#
```

Changement du groupe du fichier :

```
root@srv-nagios:/usr/lib/nagios/plugins# chgrp nagios check_ncpa.py
root@srv-nagios:/usr/lib/nagios/plugins# |
```

Vérification des permissions du fichier :

```
-rwxr-xr-x 1 nagios nagios 11937 22 nov. 2023 check_ncpa.py
root@srv-nagios:/usr/lib/nagios/plugins#
```

Modification du script check_ncpa :

```
root@srv-nagios:/usr/lib/nagios/plugins# vim check_ncpa.py
```

- Remplacée ceci:

```
#!/usr/bin/env python
```

Si nous laissons cela tel quel, le serveur Nagios ne sera pas en mesure de reconnaître le chemin.

- par cela :

```
#!/usr/bin/python3
```

Ici on a spécifié le path de la commande python.

```
root@srv-nagios:/usr/lib/nagios/plugins# which python3
/usr/bin/python3
root@srv-nagios:/usr/lib/nagios/plugins#
```

2. Test des commandes check_ncpa

La commande suivante vérifie la version de l'agent NCPA sur un serveur spécifique :

```
/usr/lib/nagios/plugins/check_ncpa.py -H 10.0.0.124 -t 'mot_de_passe' -P 5693 -M system/agent_version
```

Cette commande permet de surveiller l'utilisation du processeur sur un serveur spécifique via l'agent NCPA :

```
/usr/lib/nagios/plugins/check_ncpa.py -H 10.0.0.124 -t 'mot_de_passe' -P 5693 -M cpu/percent -w 20 -c 40 -q 'aggregate=avg'
```

Cette commande surveille l'utilisation de la mémoire virtuelle sur un serveur à l'aide de l'agent NCPA :

```
/usr/lib/nagios/plugins/check_ncpa.py -H 10.0.0.124 -t mot_de_passe -P 5693 -M memory/virtual -w 50 -c 80 -u G
```

```
root@srv-nagios:/usr/lib/nagios/plugins# /usr/lib/nagios/plugins/check_ncpa.py -H 192.168.11.50 -t 'Tokenmenu' -P 5693 -M cpu/percent -w 20 -c 40 -q 'aggregate=avg'
OK: Percent was 0.00 % | 'percent'=0.00%;20;40;
root@srv-nagios:/usr/lib/nagios/plugins# /usr/lib/nagios/plugins/check_ncpa.py -H 192.168.11.50 -t 'Tokenmenu' -P 5693 -M system/agent_version
OK: Agent_version was ['3.1.1']
root@srv-nagios:/usr/lib/nagios/plugins# /usr/lib/nagios/plugins/check_ncpa.py -H 192.168.11.50 -t 'Tokenmenu' -P 5693 -M memory/virtual -w 50 -c 80 -u G
WARNING: Memory usage was 60.20 % (Available: 3.42 GB, Total: 8.59 GB, Free: 3.42 GB, Used: 5.17 GB) | 'available'=3.42GB;;; 'total'=8.59GB;;; 'percent'=60.20%;50;80; 'free'=3.42GB;;; 'used'=5.17GB;;;
root@srv-nagios:/usr/lib/nagios/plugins#
```

3. Ajout de la config sur nagios

Configurations du fichiers commands.cfg :

```
root@srv-nagios:/usr/lib/nagios/plugins# vim /usr/local/nagios/etc/objects/commands.cfg
```

```
define command {
    command_name check_ncpa
    command_line $USER1$/check_ncpa.py -H $HOSTADDRESS$ $ARG1$
}
```

Cette commande utilise **NCPA** pour interroger un agent installé sur l'hôte cible et surveiller des services ou des ressources (tels que le CPU, la mémoire, le disque) en fonction des arguments passés.

D. Configuration du lien du client windows dans le serveur

```
root@srv-nagios:/usr/local/nagios/etc# vim /usr/local/nagios/etc/nagios.cfg
root@srv-nagios:/usr/local/nagios/etc#
```

```
# Definitions for monitoring a Windows machine
cfg_file=/usr/local/nagios/etc/objects/windows.cfg
```

```
root@srv-nagios:/usr/local/nagios/etc# vim /usr/local/nagios/etc/objects/client-windows.cfg
```

Création du fichier de configuration client-windows.cfg

```
define host {
    use windows-server ; Inherit default values from template
    host_name client1 ; The name we're giving to this host
    alias win-client ; A longer name associated with the host
    address 192.168.11.50 ; IP address of the host
    hostgroups windows-servers
}

#####
#
# HOST GROUP DEFINITIONS
#
#####

# Define a hostgroup for Windows machines
# All hosts that use the windows-server template will automatically be a member of this group

define hostgroup {
    hostgroup_name windows-servers ; The name of the hostgroup
    alias Windows Servers ; Long name of the group
}
```

Configuration du fichier de configuration client-windows.cfg

```
#cfg_file=/usr/local/nagios/etc/objects/windows.cfg
cfg_file=/usr/local/nagios/etc/objects/client-windows.cfg
```

Remarque :

En ajoutant la ligne `/usr/local/nagios/etc/client-windows.cfg` dans le fichier de configuration de Nagios, on permet la surveillance des serveurs distants, en particulier ceux qui fonctionnent sous Windows.

Ajout des services :

```
define service {
    use                generic-service
    host_name          client1
    service_description NSClient++ Version
    check_command       check_nt!CLIENTVERSION
}

# Create a service for monitoring the uptime of the server
# Change the host_name to match the name of the host you defined above

define service {
    use                generic-service
    host_name          client1
    service_description Uptime
    check_command       check_nt!UPTIME
}

# Create a service for monitoring CPU load
# Change the host_name to match the name of the host you defined above

define service {
    use                generic-service
    host_name          client1
    service_description CPU Load
    check_command       check_nt!CPULOAD!-l 5,80,90
}
```



```

# Create a service for monitoring memory usage
# Change the host_name to match the name of the host you defined above

define service {

    use                generic-service
    host_name          client1
    service_description Memory Usage
    check_command       check_nt!MEMUSE!-w 80 -c 90
}

# Create a service for monitoring C:\ disk usage
# Change the host_name to match the name of the host you defined above

define service {

    use                generic-service
    host_name          client1
    service_description C:\ Drive Space
    check_command       check_nt!USEDISKSPACE!-l c -w 80 -c 90
}

# Create a service for monitoring the W3SVC service
# Change the host_name to match the name of the host you defined above

define service {

    use                generic-service
    host_name          client1
    service_description W3SVC
    check_command       check_nt!SERVICESTATE!-d SHOWALL -l W3SVC
}

```

```

# Create a service for monitoring the Explorer.exe process
# Change the host_name to match the name of the host you defined above

define service {
    use                generic-service
    host_name          client1
    service_description Explorer
    check_command       check_nt!PROCSTATE!-d SHOWALL -l Explorer.exe
}

### define service for NCPA

define service {
    use                generic-service      ; Utilise un modèle générique pour l
e service
    host_name          client1              ; Nom d'hôte
    service_description NCPA Agent Version ; Description du service
    check_command       check_ncpa!-t 'Tokenmenu' -P 5693 -M system/agent_version
; Commande NCPA
    max_check_attempts 3
    check_interval      5
    retry_interval      1
}

define service {
    use                generic-service      ; Utilise un modèle générique pour l
e service
    host_name          client1              ; Nom d'hôte
    service_description CPU Usage           ; Description du service
    check_command       check_ncpa!-t 'Tokenmenu' -P 5693 -M cpu/percent -w 20 -c
40 -q 'aggregate=avg' ; Commande NCPA
    max_check_attempts 3
    check_interval      5
    retry_interval      1
}

```

```

define service {
    use                generic-service      ; Utilise un modèle générique pour l
e service
    host_name          client1              ; Nom d'hôte
    service_description Virtual Memory Usage ; Description du service
    check_command       check_ncpa!-t 'Tokenmenu' -P 5693 -M memory/virtual -w 50
-c 80 -u G ; Commande NCPA
    max_check_attempts 3
    check_interval      5
    retry_interval      1
}

```

Host	Status	Le
GLPI	UP	10
client1	UP	10
cit-debian-nagios	UP	10
localhost	UP	10

Results 1 - 4 of 4 Matching Hosts







Vérification de l'ajout du client windows sur l'interface web de nagios

Host	Service	Status	Last Check	Duration	Attempt	Status Information
GLPI	Local Disk	OK	10-11-2024 16:53:49	1d 2h 29m 53s	1/3	DISK OK - free space: / 270200
	PING	OK	10-11-2024 16:55:29	1d 2h 28m 38s	1/3	PING OK - Paquets perdus = 0
cit-debian-nagios	Local Disk	OK	10-11-2024 16:52:09	1d 5h 11m 21s	1/3	DISK OK - free space: / 270200
	PING	OK	10-11-2024 16:54:14	1d 5h 10m 21s	1/3	PING OK - Paquets perdus = 0
localhost	Current Load	OK	10-11-2024 16:55:54	1d 6h 23m 3s	1/4	LOAD OK - total Charge moyen
	Current Users	OK	10-11-2024 16:55:53	1d 6h 21m 48s	1/4	UTILISATEURS OK - 2 utilisate
	HTTP	OK	10-11-2024 16:54:39	1d 6h 20m 33s	1/4	HTTP OK: HTTP/1.1 200 OK -
	PING	OK	10-11-2024 16:51:19	1d 6h 19m 18s	1/4	PING OK - Paquets perdus = 0
	Root Partition	OK	10-11-2024 16:53:19	1d 6h 22m 26s	1/4	DISK OK - free space: / 270200
	SSH	OK	10-11-2024 16:55:04	1d 6h 21m 11s	1/4	SSH OK - OpenSSH_9.2p1 De
	Swap Usage	OK	10-11-2024 16:51:44	1d 6h 19m 56s	1/4	SWAP OK - 100% free (974MB
	Total Processes	OK	10-11-2024 16:52:14	1d 6h 20m 40s	1/4	PROCS OK: 0 processus avec

Results 1 - 12 of 12 Matching Services

Vérification de l'ajout des services du client windows sur l'interface web de nagios

Service Overview For All Host Groups

Linux Servers (linux-servers)				Windows Servers (windows-servers)			
Host	Status	Services	Actions	Host	Status	Services	Actions
localhost	UP	8 OK	  	client1	UP	No matching services	  

Vérification de l'ajout au groupe du client windows sur l'interface web de nagios

Problème d'affichage des service concernant le client windows :

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
GLPI	Local Disk	OK	10-11-2024 16:58:49	1d 2h 33m 3s	1/3	DISK OK - free space: / 27020MB (91% inode=96%):
	PING	OK	10-11-2024 16:55:29	1d 2h 31m 48s	1/3	PING OK - Paquets perdus = 0%, RTA = 0.46 ms
cil-debian-nagios	Local Disk	OK	10-11-2024 16:57:09	1d 5h 14m 31s	1/3	DISK OK - free space: / 27020MB (91% inode=96%):
	PING	OK	10-11-2024 16:59:14	1d 5h 13m 31s	1/3	PING OK - Paquets perdus = 0%, RTA = 1.31 ms
localhost	Current Load	OK	10-11-2024 16:55:54	1d 6h 26m 13s	1/4	LOAD OK - total Charge moyenne: 0.00, 0.00, 0.00
	Current Users	OK	10-11-2024 16:55:53	1d 6h 24m 56s	1/4	UTILISATEURS OK - 2 utilisateurs actuellement connectés sur
	HTTP	✖	10-11-2024 16:54:39	1d 6h 23m 43s	1/4	HTTP OK: HTTP/1.1 200 OK - 10975 octets en 0.001 secondes de temps de réponse
	PING	OK	10-11-2024 16:56:19	1d 6h 22m 28s	1/4	PING OK - Paquets perdus = 0%, RTA = 0.05 ms
	Root Partition	OK	10-11-2024 16:58:19	1d 6h 25m 36s	1/4	DISK OK - free space: / 27020MB (91% inode=96%):
	SSH	✖	10-11-2024 16:55:04	1d 6h 24m 21s	1/4	SSH OK - OpenSSH_9.2p1 Debian-2+deb12u3 (protocol 2.0)
	Swap Usage	OK	10-11-2024 16:56:44	1d 6h 23m 6s	1/4	SWAP OK - 100% free (974MB out of 974MB)
	Total Processes	OK	10-11-2024 16:57:14	1d 6h 23m 50s	1/4	PROCS OK: 0 processus avec ETAT = RSZDT

Results 1 - 12 of 12 Matching Services

Actuellement, je rencontre un problème d'affichage avec le service que j'ai configuré dans le fichier de configuration de Nagios. Bien que le client Windows soit correctement visible dans la liste des hôtes, le service lui-même ne s'affiche pas. Cela démontre que j'ai réussi à installer et à monter le client avec succès.

J'ai également vérifié les plugins installés et les commandes que j'utilise, et j'obtiens des résultats positifs dans le terminal. Cela me permet de conclure que le problème ne réside pas dans les vérifications elles-mêmes, mais plutôt dans un autre aspect que je n'ai pas pu identifier, faute de temps.

Commande de vérification :

```
/usr/lib/nagios/plugins/check_ncpa.py -H 192.168.11.50 -t 'Tokenmenu' -P 5693 -M system/agent_version
```

```
/usr/lib/nagios/plugins/check_ncpa.py -H 192.168.11.50 -t 'Tokenmenu' -P 5693 -M cpu/percent -w 20 -c 40 -q 'aggregate=avg'
```

```
/usr/lib/nagios/plugins/check_ncpa.py -H 192.168.11.50 -t 'Tokenmenu' -P 5693 -M memory/virtual -w 50 -c 80 -u G
```

Résultat obtenus :

```
root@srv-nagios:/usr/lib/nagios/plugins# /usr/lib/nagios/plugins/check_ncpa.py -H 192.168.11.50 -t 'Tokenmenu' -P 5693 -M cpu/percent -w 20 -c 40 -q 'aggregate=avg'
OK: Percent was 0.00 % | 'percent'=0.00%;20;40;
root@srv-nagios:/usr/lib/nagios/plugins# /usr/lib/nagios/plugins/check_ncpa.py -H 192.168.11.50 -t 'Tokenmenu' -P 5693 -M system/agent_version
OK: Agent_version was ['3.1.1']
root@srv-nagios:/usr/lib/nagios/plugins# /usr/lib/nagios/plugins/check_ncpa.py -H 192.168.11.50 -t 'Tokenmenu' -P 5693 -M memory/virtual -w 50 -c 80 -u G
WARNING: Memory usage was 60.20 % (Available: 3.42 GB, Total: 8.59 GB, Free: 3.42 GB, Used: 5.17 GB) | 'available'=3.42GB;;; 'total'=8.59GB;;; 'percent'=60.20%;50;80; 'free'=3.42GB;;; 'used'=5.17GB;;;
root@srv-nagios:/usr/lib/nagios/plugins# █
```

Bien que j'utilise actuellement le plugin **check_ncpa** pour surveiller les services, j'aurais également pu opter pour **NSClient++** comme alternative. NSClient++ est un agent de surveillance conçu spécifiquement pour les environnements Windows. Il offre une intégration fluide avec Nagios et permet de surveiller divers aspects du système, tels que l'utilisation de la mémoire, l'espace disque et les processus, tout en offrant des fonctionnalités de notification et d'alerte.

L'un des avantages d'utiliser NSClient++ est sa flexibilité et sa simplicité de configuration, ce qui peut faciliter la gestion des hôtes Windows.

E. Modification de l'utilisateur admin

Pour une meilleure gestion des utilisateurs, nous allons remplacer l'utilisateur **admin** par un nom plus explicite, ici **nagiosadmin** :

```
root@srv-nagios:~# vim /usr/local/nagios/etc/htpasswd.users
```

```
nagiosadmin:$apr1$3lI4yz4b$1k1DJCI./o2hLp/E5yWhT0
```

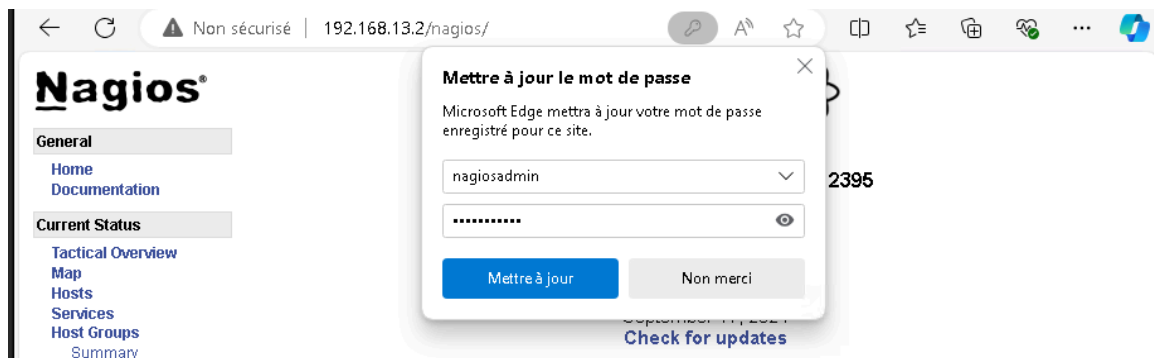
MAJ du mdp :

```
root@srv-nagios:~/nagios-4.5.5# htpasswd /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Updating password for user nagiosadmin
```

Nouveau login :

login : nagiosadmin

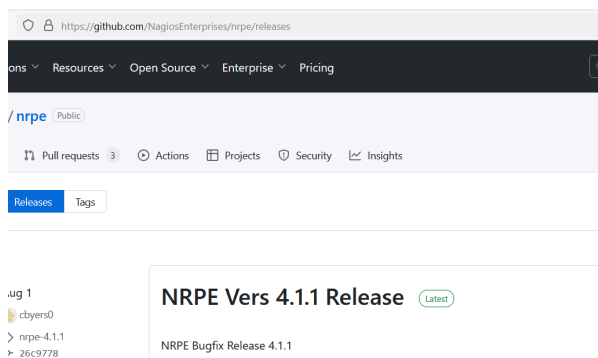
mdp : nagiosamdin



F. Supervision sur un client linux

1. Installation et configuration de NRPE (Serveur nagios)

On va installer la dernière version du plugin :



```
root@srv-nagios:~# cd /tmp
root@srv-nagios:/tmp# wget --no-check-certificate -O nrpe.tar.gz https://github.com/NagiosEnterprises/nrpe/releases/download/nrpe-4.1.1/nrpe-4.1.1.tar.gz
```

Lien du paquet installé :

<https://github.com/NagiosEnterprises/nrpe/releases/download/nrpe-4.1.1/nrpe-4.1.1.tar.gz>

```
root@srv-nagios:/tmp# tar xzf nrpe.tar.gz
```

Compilation du code source de NRPE :

```
root@srv-nagios:/tmp/nrpe-4.1.1# ./configure --enable-command-args --with-ssl-lib=/usr/lib/x86_64-linux-gnu/
```

Configuration terminée :

```
*** Configuration summary for nrpe 4.1.1 2024-08-01 ***:
```

```
General Options:
```

```
-----
NRPE port:      5666
NRPE user:      nagios
NRPE group:     nagios
Nagios user:    nagios
Nagios group:   nagios
```

```
Review the options above for accuracy.  If they look okay,
type 'make all' to compile the NRPE daemon and client
or type 'make' to get a list of make options.
```

```
root@srv-nagios:/tmp/nrpe-4.1.1# |
```

Installation des commandes ci dessus :

```
sudo make all
sudo make install-groups-users
sudo make install
sudo make install-config
```

```
root@srv-nagios:/tmp/nrpe-4.1.1# make all
cd ./src/; make
make[1] : on entre dans le répertoire « /tmp/nrpe-4.1.1/src »
gcc -g -O2 -I/usr/include/openssl -DHAVE_CONFIG_H -I ../include -I ../include -o nrpe ./nrpe.c ./u
tils.c ./acl.c -L/usr/lib/x86_64-linux-gnu -lssl -lcrypto -lnsl
gcc -g -O2 -I/usr/include/openssl -DHAVE_CONFIG_H -I ../include -I ../include -o check_nrpe ./chec
k_nrpe.c ./utils.c -L/usr/lib/x86_64-linux-gnu -lssl -lcrypto -lnsl
make[1] : on quitte le répertoire « /tmp/nrpe-4.1.1/src »

*** Compile finished ***

You can now continue with the installation or upgrade process.

Read the PDF documentation (docs/NRPE.pdf) for information on the next
steps you should take to complete the installation or upgrade.
```

```
root@srv-nagios:/tmp/nrpe-4.1.1# make install-groups-users
Group nagios already exists
User nagios already exists
root@srv-nagios:/tmp/nrpe-4.1.1#
```

```
cd ./src/; make install
make[1] : on entre dans le répertoire « /tmp/nrpe-4.1.1/src »
make install-plugin
make[2] : on entre dans le répertoire « /tmp/nrpe-4.1.1/src »
/usr/bin/install -c -m 755 -d /usr/local/nagios/bin
/usr/bin/install -c -m 755 ../uninstall /usr/local/nagios/bin/nrpe-uninstall
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/libexec
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/libexec
/usr/bin/install -c -m 775 -o nagios -g nagios check_nrpe /usr/local/nagios/libexec
make[2] : on quitte le répertoire « /tmp/nrpe-4.1.1/src »
make install-daemon
make[2] : on entre dans le répertoire « /tmp/nrpe-4.1.1/src »
/usr/bin/install -c -m 755 -d /usr/local/nagios/bin
/usr/bin/install -c -m 755 ../uninstall /usr/local/nagios/bin/nrpe-uninstall
/usr/bin/install -c -m 755 nrpe /usr/local/nagios/bin
/usr/bin/install -c -m 755 -d /usr/lib/tmpfiles.d
/usr/bin/install -c -m 644 ../startup/tmpfile.conf /usr/lib/tmpfiles.d/nrpe.conf
make[2] : on quitte le répertoire « /tmp/nrpe-4.1.1/src »
make[1] : on quitte le répertoire « /tmp/nrpe-4.1.1/src »
```



```
root@srv-nagios:/tmp/nrpe-4.1.1# make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 644 -o nagios -g nagios sample-config/nrpe.cfg /usr/local/nagios/etc
root@srv-nagios:/tmp/nrpe-4.1.1#
```

Mise à jour des fichiers de service pour lier les services de Nagios au port TCP 5666 (port par défaut de NRPE) :

```
sudo sh -c "echo >> /etc/services"
sudo sh -c "sudo echo '# Nagios services' >> /etc/services"
sudo sh -c "sudo echo 'nrpe 5666/tcp' >> /etc/services"
```

Cela permettra au système de reconnaître que le service NRPE utilise le port **5666**. Dans notre cas, les commandes automatiques ne fonctionnaient pas, nous avons donc dû effectuer cette modification manuellement.

Pour ce faire, vérifiez dans le fichier **/etc/services** si le port **5666** est déjà associé à NRPE. Si ce n'est pas le cas, ajoutez-le :

```
# Nagios services
nrpe 5666/tcp
```

Installation et configuration du service daemon (NRPE) :

```
sudo make install-init
sudo systemctl enable nrpe.service
```

```
root@srv-nagios:/tmp/nrpe-4.1.1# make install-init
/usr/bin/install -c -m 644 startup/default-service /lib/systemd/system/nrpe.service
root@srv-nagios:/tmp/nrpe-4.1.1# systemctl enable nrpe.service
Created symlink /etc/systemd/system/multi-user.target.wants/nrpe.service → /lib/systemd/system/nrpe.service.
root@srv-nagios:/tmp/nrpe-4.1.1#
```

Ajout de l'ip du serveur nagios dans nrpe.cfg :

```
root@srv-nagios:/tmp/nrpe-4.1.1# vim /usr/local/nagios/etc/nrpe.cfg
```

```
allowed_hosts=127.0.0.1,192.168.13.2
```

Nous voulons que la valeur dans dont_blame_nrpe=0 soit changée en dont_blame_nrpe=1.

```
dont_blame_nrpe=1
```

Vérification que les commandes ci-dessus ne sont pas commentées :

```
command[check_users]=/usr/lib/nagios/plugins/check_users -w 5 -c 10
command[check_load]=/usr/lib/nagios/plugins/check_load -w 15,10,5 -c 30,25,20
command[check_hda1]=/usr/lib/nagios/plugins/check_disk -w 20% -c 10% -p /dev/hda1
command[check_zombie_procs]=/usr/lib/nagios/plugins/check_procs -w 5 -c 10 -s Z
command[check_total_procs]=/usr/lib/nagios/plugins/check_procs -w 150 -c 200
```

```
root@srv-nagios:/tmp/nrpe-4.1.1# vim /etc/nagios/nrpe.cfg
```

```
command[check_users]=/usr/lib/nagios/plugins/check_users -w 5 -c 10
command[check_load]=/usr/lib/nagios/plugins/check_load -r -w .15,.10,.05 -c .30,.25,.20
command[check_hda1]=/usr/lib/nagios/plugins/check_disk -w 20% -c 10% -p /dev/hda1
command[check_zombie_procs]=/usr/lib/nagios/plugins/check_procs -w 5 -c 10 -s Z
command[check_total_procs]=/usr/lib/nagios/plugins/check_procs -w 150 -c 200
```

Installation et lancement le daemon NRPE :

```
root@srv-nagios:~# apt install nagios-nrpe-server
```

```
root@srv-nagios:~# /etc/init.d/nagios-nrpe-server start
Starting nagios-nrpe-server (via systemctl): nagios-nrpe-server.service.
root@srv-nagios:~#
```

```

root@srv-nagios:~# systemctl status nagios-nrpe-server.service
● nagios-nrpe-server.service - Nagios Remote Plugin Executor
   Loaded: loaded (/lib/systemd/system/nagios-nrpe-server.service; enabled; preset: enabled)
   Active: active (running) since Wed 2024-10-09 22:54:58 CEST; 1min 8s ago
     Docs: http://www.nagios.org/documentation
    Main PID: 4441 (nrpe)
      Tasks: 1 (limit: 4652)
     Memory: 1.2M
        CPU: 6ms
    CGroup: /system.slice/nagios-nrpe-server.service
            └─4441 /usr/sbin/nrpe -c /etc/nagios/nrpe.cfg -f

oct. 09 22:54:58 srv-nagios systemd[1]: Started nagios-nrpe-server.service - Nagios Remote Plugin Executor.
oct. 09 22:54:58 srv-nagios nrpe[4441]: Starting up daemon
oct. 09 22:54:58 srv-nagios nrpe[4441]: Server listening on 0.0.0.0 port 5666.
oct. 09 22:54:58 srv-nagios nrpe[4441]: Server listening on :: port 5666.
oct. 09 22:54:58 srv-nagios nrpe[4441]: Listening for connections on port 5666
oct. 09 22:54:58 srv-nagios nrpe[4441]: Allowing connections from: 127.0.0.1,::1
root@srv-nagios:~#

```

Pour que le service démarre automatiquement au démarrage de la machine :

```

root@srv-nagios:~# systemctl enable nagios-nrpe-server

```

```

root@srv-nagios:/tmp/nrpe-4.1.1# systemctl enable nagios-nrpe-server
Synchronizing state of nagios-nrpe-server.service with SysV service script with /lib/systemd/systemd
-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable nagios-nrpe-server
root@srv-nagios:/tmp/nrpe-4.1.1#

```

G. Étapes pour installer et configurer NRPE sur l'hôte client Debian

Commande utilisée :

apt update

apt install nagios-nrpe-server nagios-plugins

```
root@debian:~# apt update
Atteint :1 http://security.debian.org/debian-security bookworm-security InRelease
Atteint :2 http://deb.debian.org/debian bookworm InRelease
Atteint :3 http://deb.debian.org/debian bookworm-updates InRelease
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Tous les paquets sont à jour.
root@debian:~# apt install nagios-nrpe-server nagios-plugins
```

```
root@clt-debian-nagios:~# apt install nagios-nrpe-server nagios-plugins
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Note : sélection de « monitoring-plugins » au lieu de « nagios-plugins »
Les paquets supplémentaires suivants seront installés :
  libdb11 libmariadb3 libnet-snmp-perl libpq5 libradcli4 liburiparser1 mariadb-common
  monitoring-plugins-basic monitoring-plugins-common monitoring-plugins-standard mysql-common
  python3-gpg python3-ldb python3-samba python3-talloc python3-tdb rpcbind samba-common
  samba-common-bin samba-dsdb-modules smbclient snmp
Paquets suggérés :
  libcrypt-des-perl libdigest-hmac-perl libio-socket-inet6-perl icinga2 nagios-plugins-contrib
  fping postfix | sendmail-bin | exim4-daemon-heavy | exim4-daemon-light qstat xinetd | inetd
  heimdal-clients python3-markdown python3-dnspython cifs-utils
Les NOUVEAUX paquets suivants seront installés :
  libdb11 libmariadb3 libnet-snmp-perl libpq5 libradcli4 liburiparser1 mariadb-common
  monitoring-plugins monitoring-plugins-basic monitoring-plugins-common
  monitoring-plugins-standard mysql-common nagios-nrpe-server python3-gpg python3-ldb
  python3-samba python3-talloc python3-tdb rpcbind samba-common samba-common-bin
  samba-dsdb-modules smbclient snmp
0 mis à jour, 24 nouvellement installés, 0 à enlever et 41 non mis à jour.
Il est nécessaire de prendre 7 003 ko dans les archives.
Après cette opération, 39,6 Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] O
Réception de :1 http://deb.debian.org/debian bookworm/main amd64 nagios-nrpe-server amd64 4.1.0-1+b1
[367 kB]
Réception de :2 http://deb.debian.org/debian bookworm/main amd64 rpcbind amd64 1.2.6-6+b1 [48,3 kB]
Réception de :3 http://deb.debian.org/debian bookworm/main amd64 libdb11 amd64 0.9.0-6 [28,9 kB]
Réception de :4 http://deb.debian.org/debian bookworm/main amd64 mysql-common all 5.8+1.1.0 [6 636 B]
Réception de :5 http://deb.debian.org/debian bookworm/main amd64 mariadb-common all 1:10.11.6-0+deb1
2u1 [24,4 kB]
Réception de :6 http://deb.debian.org/debian bookworm/main amd64 libmariadb3 amd64 1:10.11.6-0+deb12
u1 [174 kB]
Réception de :7 http://deb.debian.org/debian bookworm/main amd64 libnet-snmp-perl all 6.0.1-7 [94,7
```

```
root@debian:~# systemctl restart nagios-nrpe-server
root@debian:~#
```

Vérification de la connexion NRPE depuis le serveur Nagios :

Commande utilisée :

/usr/local/nagios/libexec/check_nrpe -H <ip_machine>

```
root@srv-nagios:/tmp/nagios-plugins-2.4.12# /usr/local/nagios/libexec/check_nrpe -H 192.168.1.182
NRPE v4.1.0
root@srv-nagios:/tmp/nagios-plugins-2.4.12#
```

La réponse **NRPE v4.1.0** indique que le service NRPE est correctement installé et fonctionne sur la machine distante.

H. Étapes pour Surveiller les hôtes Linux sur le serveur

Après l'installation de **Nagios** et de **NRPE**, il est nécessaire de rendre l'hôte distant visible pour le serveur Nagios afin de pouvoir surveiller ses services. Cette opération consiste à créer un fichier de configuration spécifique pour chaque hôte, situé dans le répertoire **/usr/local/nagios/etc/servers**.

Le fichier de configuration définit les services à surveiller pour cet hôte particulier. Il est recommandé de créer des fichiers séparés pour chaque hôte plutôt que d'utiliser un seul fichier pour tous, afin de maintenir une gestion plus claire et modulable des hôtes et de leurs services.

Ici on a crée **/usr/local/nagios/etc/servers/debian-client.cfg**

```
root@srv-nagios:/tmp/nagios-plugins-2.4.12# ls /usr/local/nagios/etc/servers
root@srv-nagios:/tmp/nagios-plugins-2.4.12# vim /usr/local/nagios/etc/servers/debian-client.cfg
```

1. Configuration des templates

On va voir les templates disponible pour configurer les hôtes et les services à surveiller :

```
root@srv-nagios:~# vim /usr/local/nagios/etc/objects/templates.cfg
```

```
define host {
    name                linux-server          ; The name of this host template
    use                 generic-host          ; This template inherits other values fr
    ; From the generic-host template
    check_period        24x7                  ; By default, Linux hosts are checked ro
    ; Around the clock
    check_interval      5                     ; Actively check the host every 5 minute
    ; Seconds
    retry_interval      1                     ; Schedule host check retries at 1 minut
    ; e intervals
    max_check_attempts  10                    ; Check each Linux host 10 times (max)
    check_command        check-host-alive     ; Default command to check Linux hosts
    notification_period workhours             ; Linux admins hate to be woken up, so w
    ; e only notify during the day
    ; Note that the notification_period vari
    ; able is being overridden from
    ; the value that is inherited from the g
    ;eneric-host template!
    notification_interval 120                 ; Resend notifications every 2 hours
    notification_options d,u,r               ; Only send notifications for specific h
    ;ost states
    contact_groups       admins               ; Notifications get sent to the admins b
    ;y default
    register              0                   ; DON'T REGISTER THIS DEFINITION - ITS N
    ; NOT A REAL HOST, JUST A TEMPLATE!
}
```

Le bloc **define host** est essentiel dans Nagios car il permet de définir chaque hôte que le système doit surveiller, en précisant des informations spécifiques comme le nom de l'hôte, son adresse IP, et les services à vérifier. Bien que des paramètres généraux puissent être définis dans un template pour simplifier la configuration, chaque hôte doit avoir son propre fichier de configuration pour personnaliser ses caractéristiques. Par exemple, dans un fichier comme **debian-client.cfg**, il est nécessaire de redéfinir un **define host** pour spécifier l'adresse IP du serveur et tout autre détail propre à cet hôte. Cela permet à Nagios de savoir précisément quel serveur surveiller et comment procéder, tout en héritant des

configurations communes définies dans les templates. Ainsi, chaque machine bénéficie d'une configuration personnalisée tout en profitant de la simplicité et de la flexibilité apportées par les templates.

I. Configuration de l'hôte dans client-debian.cfg

Configuration de l'hôte dans client-debian.cfg :

Le fichier **client-debian.cfg** permet au serveur Nagios de recevoir les informations de l'hôte Debian. Il définit l'adresse IP, le nom de l'hôte et les services à surveiller, assurant ainsi la communication entre le serveur et l'hôte pour une surveillance efficace.

```
# Nagios Host configuration file template
define host {
    use                linux-server
    host_name          clt-debian-nagios
    alias              Debian Cilent Nagios
    address             192.168.11.17
    register            1
}
```

Pour vérifier si il y a les erreurs :

```
root@srv-nagios:/tmp/nagios-plugins-2.4.12# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

Redémarrage des services et relance du NRPE sur le server nagios :

```
root@srv-nagios:~# service apache2 restart
root@srv-nagios:~# service nagios restart
root@srv-nagios:~#
```

```

Nagios Core 4.5.5
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-09-17
License: GPL

Website: https://www.nagios.org
Reading configuration data...
  Read main config file okay...
  Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
  Checked 8 services.
  Checked 2 hosts.
  Checked 1 host groups.
  Checked 0 service groups.
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 24 commands.
  Checked 5 time periods.
  Checked 0 host escalations.
  Checked 0 service escalations.
Checking for circular paths...
  Checked 2 hosts
  Checked 0 service dependencies
  Checked 0 host dependencies
  Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...







Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
root@srv-nagios:~#

```

Vérification de la présence de notre client Nagios dans le menu :

Limit Results:

Host 	Status 	Last Check 	Duration 	Status Information
cit-debian-nagios 	UP	10-10-2024 10:42:10	0d 0h 12m 17s	PING OK - Paquets perdus = 0%,
localhost 	UP	10-10-2024 10:43:21	0d 0h 11m 6s	PING OK - Paquets perdus = 0%,

Results 1 - 2 of 2 Matching Hosts

J. Intégration et configuration des templates services

Intégration des templates services pour le client :

vim /usr/local/nagios/etc/objects/templates.cfg

Les nouveaux ajouts permettent de définir les éléments à surveiller sur notre client :

```
#####
#
# SERVICE TEMPLATES
#####

# Generic service definition template
# This is NOT a real service, just a template!

define service {
    name generic-service ; The 'name' of this service
    active_checks_enabled 1 ; Active service checks
    passive_checks_enabled 1 ; Passive service checks
    parallelize_check 1 ; Active service checks
    obsess_over_service 1 ; We should obsess over this service (if needed)
    check_freshness 0 ; Default is to NOT check freshness
    notifications_enabled 1 ; Service notifications
    event_handler_enabled 1 ; Service event handler
    flap_detection_enabled 1 ; Flap detection is enabled
    process_perf_data 1 ; Process performance data
    retain_status_information 1 ; Retain status information
    retain_nonstatus_information 1 ; Retain non-status information
    is_volatile 0 ; The service is not volatile
    check_period 24x7 ; The service can be checked at any time
    max_check_attempts 3 ; Re-check the service 3 times
    check_interval 10 ; Check the service every 10 seconds
    retry_interval 2 ; Re-check the service 2 seconds after a failed attempt
    contact_groups admins ; Notifications get sent to the admins group when the service is down
}
```

Configuration des Services à Superviser :

Pour superviser efficacement ces services, il est essentiel de les configurer dans les templates en incluant les éléments suivants :

- L'état de fonctionnement (allumé/éteint),
- La charge du système,

- L'état des interfaces réseau (up/down),
- Pour les ordinateurs, le taux de remplissage du disque dur.

Exemple de service ajoutée :

```
define service {
    use                generic-monitoring-service
    host_name          mtr-ubuntu
    service_description PING
    check_command       check_ping!100.0,20%!500.0,60%
}
```

```
define service {
    use                generic-load-service
    host_name          mtr-ubuntu
    service_description Load Average
    check_command       check_load!15,10,5!20,15,10
}
```

```
define service {
    use                generic-disk-service
    host_name          mtr-ubuntu
    service_description Local Disk
    check_command       check_local_disk!20%!10%!/
}
```

```

# Services pour superviser (AP5)

define service {
    name                generic-monitoring-service
    check_interval       5
    retry_interval       1
    max_check_attempts   3
    check_period         24x7
    notification_interval 60
    notification_period   24x7
    notifications_enabled 1
    contact_groups        admins
    register              0
}

define service {
    use                  generic-monitoring-service
    host_name            nom_de_l_hote
    service_description   Host State (On/Off)
    check_command         check-host-alive
}

define service {
    use                  generic-monitoring-service
    host_name            nom_de_l_hote
    service_description   System Load
    check_command         check_load!5,10,15!10,20,25
}

define service {
    use                  generic-monitoring-service
    host_name            nom_de_l_hote
    service_description   Network Interface Status
    check_command         check_ifoperstatus!eth0!1!1
}

define service {
    use                  generic-monitoring-service
    host_name            nom_de_l_hote
    service_description   Disk Space Usage
    check_command         check_local_disk!20%!10%!/
}

```

Visualisation des templates ajoutés

1. Installation des plugins nécessaire pour la supervision

Chaque service utilise des plugins présents dans notre dossier plugins. Nous allons donc copier tout son contenu et le déplacer dans le répertoire libexec, afin de garantir que Nagios puisse accéder et exécuter les plugins nécessaires pour surveiller les différents services.

```
root@srv-nagios:~# cp /usr/lib/nagios/plugins/* /usr/local/nagios/libexec/  
root@srv-nagios:~#
```

Ici, nous copions tous les plugins du répertoire **plugins** vers le répertoire **libexec**. Ensuite, nous ajoutons les templates nécessaires dans le fichier cpng.cfg. Enfin, nous redémarrons les services apache2 et nagios pour appliquer les modifications et assurer le bon fonctionnement de la surveillance.

K. Intégration du Service GLPI dans Nagios

```
root@srv-glpi:~# apt install nagios-nrpe-server nagios-plugins
```

```
root@srv-glpi:~# vim /etc/nagios/nrpe.cfg
```

```
# Note: This option is ignored if using SSL  
allowed_hosts=127.0.0.1,192.168.13.2
```

Autoriser l'accès depuis le serveur avec l'IP spécifiée

```
root@srv-nagios:~# /usr/local/nagios/libexec/check_nrpe -H 192.168.13.1  
NRPE v4.1.0  
root@srv-nagios:~#
```

```
root@srv-nagios:~# vim /usr/local/nagios/etc/servers/srv-glpi.cfg
```

```
# Nagios Host configuration file template
define host {
    use                linux-server
    host_name          GLPI
    alias              GLPI server
    address            192.168.13.1
    register            1
}
```

Configuration de l'hôte

```
root@srv-nagios:~# vim /usr/local/nagios/etc/servers/srv-glpi.cfg
root@srv-nagios:~# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

```
root@srv-nagios:~# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.5.5
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-09-17
License: GPL

Website: https://www.nagios.org
Reading configuration data...
    Read main config file okay...
    Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
    Checked 10 services.
    Checked 3 hosts.
    Checked 1 host groups.
    Checked 0 service groups.
    Checked 1 contacts.
    Checked 1 contact groups.
    Checked 24 commands.
    Checked 5 time periods.
    Checked 0 host escalations.
    Checked 0 service escalations.
Checking for circular paths...
    Checked 3 hosts
    Checked 0 service dependencies
    Checked 0 host dependencies
    Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0
```

```
# Nagios Host configuration file template
define host {
    use                linux-server
    host_name          GLPI
    alias              GLPI server
    address            192.168.13.1
    register           1
}

define service {
    use                generic-monitoring-service
    host_name          GLPI
    service_description PING
    check_command       check_ping!100.0,20%!500.0,60%
}

define service {
    use                generic-load-service
    host_name          GLPI
    service_description Load Average
    check_command       check_load!15,10,5!20,15,10
}

define service {
    use                generic-disk-service
    host_name          GLPI
    service_description Local Disk
    check_command       check_local_disk!20%!10%!/
}
```

Configuration de des services à surveiller

Host ↕	Status ↕	Last Check ↕	Duration ↕	Status Information
GLPI	UP	10-10-2024 14:20:41	0d 0h 19m 53s	PING OK - Paquets perdus = 0%, RTA = 0.45 ms
clt-debian-nagios	UP	10-10-2024 14:24:07	0d 3h 53m 24s	PING OK - Paquets perdus = 0%, RTA = 1.14 ms
localhost	UP	10-10-2024 14:21:33	0d 3h 52m 13s	PING OK - Paquets perdus = 0%, RTA = 0.06 ms

Affichage de l'Intégration du Serveur GLPI dans l'Interface Web de Nagios

VI. MISSION 2 : Sauvegarde d'actifs réseaux avec Rancid

Expliquer en quelques lignes le fonctionnement de RANCID et les avantages de cette solution pour Menuimétal :

Rancid (***Really Awesome New Cisco config Differ***) est un outil de gestion des configurations réseau qui permet de sauvegarder et de modifier les configurations des routeurs et les switchs. Il est basé sur les protocoles comme Telnet et SSH.

Cette solution présente un avantage pour Menuimetal ; cela permet à la société d'automatiser les sauvegardes des configurations des équipements réseau ainsi que le nettoyage des logs.

En cas de problème, on peut ainsi aisément déployer une backup.

Installation de Rancid :

```
apt install rancid
```

```
apt install cvs cvsweb expect
```

```
root@srv-rancid:/var/lib/rancid/bin# apt install cvs cvsweb expect
```

Configuration des groupes :

```
# list of rancid groups
LIST_OF_GROUPS="routeurs switchs"; export LIST_OF_GROUPS
# more groups...
#LIST_OF_GROUPS="$LIST_OF_GROUPS noc billybobisp"; export LIST_OF_GROUPS
#
```

Chemin : **/etc/rancid/rancid.conf**

En exécutant la commande **rancid-cvs**, rancid crée les dossiers correspondant aux groupes :

```
rancid@srv-rancid:~$ rancid-cvs

No conflicts created by this import

cvs checkout: Updating routeurs
Directory /var/lib/rancid/CVS/routeurs/configs put under version control
cvs commit: Examining configs
cvs add: scheduling file `router.db' for addition
cvs add: use `cvs commit' to add this file permanently
/var/lib/rancid/CVS/routeurs/router.db,v <-- router.db
initial revision: 1.1

No conflicts created by this import

cvs checkout: Updating switches
Directory /var/lib/rancid/CVS/switchs/configs put under version control
cvs commit: Examining configs
cvs add: scheduling file `router.db' for addition
cvs add: use `cvs commit' to add this file permanently
/var/lib/rancid/CVS/switchs/router.db,v <-- router.db
initial revision: 1.1
```

```
bin downreport getipacctg logs rancid-cvspurge routeurs rtrfilter switchs
```

Ajout de l'utilisateur rancid :

```
root@srv-rancid:/var/lib/rancid/bin# adduser rancid --home /home/rancid
```

Lancement du service via l'utilisateur rancid :

su - rancid pour se connecter en tant qu'utilisateur rancid

rancid-run pour lancer rancid

Configuration du fichier router.db qui regroupe toutes les informations du switch :

```
192.168.13.60:HP:up
```

Création du fichier `/var/lib/rancid/.cloginrc` qui sert à gérer l'authentification sur le switch :

```
#Switch HP  
add method 192.168.13.60 telnet
```

A. Installation de TFTP

```
root@srv-rancid:/var/lib/rancid# apt install tftpd-hpa
```

Création du répertoire tftpboot :

```
root@srv-rancid:/var/lib# mkdir tftpboot
```

Configuration du fichier `/var/lib/tftpboot` :

```
192.168.13.3 x  
# /etc/default/tftpd-hpa  
TFTP_USERNAME="tftp"  
TFTP_DIRECTORY="/var/lib/tftpboot"  
TFTP_ADDRESS=":69"  
TFTP_OPTIONS="--secure"
```

```

root@srv-rancid:/var/lib/tftpboot# systemctl status tftpd-hpa
● tftpd-hpa.service - LSB: HPA's tftp server
   Loaded: loaded (/etc/init.d/tftpd-hpa; generated)
   Active: active (running) since Thu 2024-10-10 08:47:12 CEST; 5min ago
     Docs: man:systemd-sysv-generator(8)
  Process: 2921 ExecStart=/etc/init.d/tftpd-hpa start (code=exited, status=0/SUCCESS)
    Tasks: 1 (limit: 2306)
   Memory: 416.0K
      CPU: 19ms
   CGroup: /system.slice/tftpd-hpa.service
           └─2930 /usr/sbin/in.tftpd --listen --user tftp --address :69 --secure /var/lib/tftpboot

```

B. Configuration du switch HP et récupération de la config via TFTP

Suppression de la configuration existante :

configure terminal

erase startup-config

Configuration du VLAN, des ports et de l'IP :

```

ProCurve Switch 2626# configure terminal
ProCurve Switch 2626(config)# vlan 410
ProCurve Switch 2626(vlan-410)# untagged 5-6
ProCurve Switch 2626(vlan-410)# ip address 192.168.13.50 255.255.255.0

```

name gestion pour configurer le nom du VLAN


Configuration de tous les VLANs du switch (sh run) :

```
vlan 1
 name "DEFAULT_VLAN"
 untagged 15-20,25-26
 ip address dhcp-bootp
 no untagged 1-14,21-24
 exit
vlan 410
 name "gestion"
 untagged 5-6
 ip address 192.168.13.60 255.255.255.0
 exit
vlan 411
 name "lan"
 untagged 7-8
 no ip address
 exit
vlan 412
 name "dnz"
 untagged 9-10
 no ip address
 exit
vlan 413
 name "internet"
 untagged 11-12
 no ip address
 exit
vlan 414
 name "invites"
 untagged 13-14
 no ip address
 exit
vlan 50
 name "VLAN50"
 untagged 1-4,21-24
 exit
vlan 99
 name "VLAN99"
 exit
```

Test de connexion avec le switch dans le Vlan Gestion :

```
root@C419-11:~# ping 192.168.13.60
PING 192.168.13.60 (192.168.13.60) 56(84) bytes of data.
 64 bytes from 192.168.13.60: icmp_seq=1 ttl=64 time=15.4 ms
 64 bytes from 192.168.13.60: icmp_seq=2 ttl=64 time=13.9 ms
 64 bytes from 192.168.13.60: icmp_seq=3 ttl=64 time=11.9 ms
 64 bytes from 192.168.13.60: icmp_seq=4 ttl=64 time=10.4 ms
 64 bytes from 192.168.13.60: icmp_seq=5 ttl=64 time=8.89 ms
```

Récupération de la config du switch via TFTP :

 *Le fichier que l'on indique pour récupérer la configuration du switch doit être au préalable créé sur le serveur.*

*Le dossier tftpboot ainsi que ce fichier doivent avoir les droits en lecture, écriture et exécution : **chmod -R 777 /var/lib/tftpboot***

```
ProCurve Switch 2626# copy startup-config tftp 192.168.13.3 config.txt
```

copy startup-config tftp 192.168.13.3 config.txt

Vérification sur notre serveur TFTP :

```
J4900B Configuration Editor; Created on release #H.10.83

hostname "ProCurve Switch 2626"
snmp-server community "public" Unrestricted
vlan 1
    name "DEFAULT_VLAN"
    untagged 15-20,25-26
    ip address dhcp-bootp
    no untagged 1-14,21-24
    exit
vlan 410
    name "gestion"
    untagged 5-6
    ip address 192.168.13.60 255.255.255.0
    exit
vlan 411
    name "lan"
    untagged 7-8
    no ip address
    exit
vlan 412
    name "dmz"
    untagged 9-10
    no ip address
    exit
vlan 413
    name "internet"
    untagged 11-12
    no ip address
    exit
vlan 414
    name "invites"
    untagged 13-14
    no ip address
    exit
vlan 50
    name "VLAN50"
    untagged 1-4,21-24
    exit
vlan 99
    name "VLAN99"
    exit
```