

Compte rendu sujet 11 :

NAGIOS

Installation et configuration

Travail encadré par : Mme. ZAYDI Mouia

Réalisé par :

- **LHOUNBRI Soukaina**
- **FARAH Hafsa**
- **ELFADIL Asmaa**
- **AMSMAR Lamyae**
- **SALKA Fatima**
- **ID-OUAARAB Khalid**
- **MANSOUR Yassine**

Introduction

Certes, les entreprises connaissent l'intervention plusieurs types d'éléments : les être-humaines, les équipements physiques et matériels ainsi que des flux logiques ou énergétiques. Ceci fait que la détection de problèmes au niveau du système devient très difficile, surtout lorsque ces problèmes ne résultent pas d'erreurs explicites au niveau du système. Par conséquent, on risque d'accumuler des 'petites' erreurs qui causeraient sûrement dans le futur des failles qui sont graves et parfois irréversibles. Pour éviter ce genre de problème, on a recours à la supervision.

La supervision est une technique industrielle de suivi et de pilotage informatique qui permet aux administrateurs la surveillance, la visualisation et l'analyse de l'état de santé des systèmes d'informations, en outre de se faire remonter toutes les anomalies ou les interruptions qui feraient que ce système d'information pourrait à un instant s'arrête.

On distingue différents types de supervision : La supervision technique qui permet de surveiller le réseau, l'infrastructure et les machines de système d'information. La supervision applicative qui va consister à surveiller les applications et les processus métiers. Le contrat de service

qui permet de surveiller le respect des indicateurs. La supervision Métier qui va consister à surveiller les processus métiers de l'entreprise. En résumé, la supervision dépend de l'activité de l'entreprise et de son besoin.

La supervision permet de visualiser notre système d'information pour avoir une image complète sur son état. Elle nous permet de détecter les erreurs et même de les anticiper d'une manière rapide et efficace. L'utilisation d'un outil de supervision s'avère donc être indispensable pour superviser techniquement, pour agir en cas de nécessité ainsi que pour mettre à jour les données de supervision en cas d'une modification un ajout ou élimination d'un élément du système.

Après avoir été conscient de la nécessité de mettre en place une solution de supervision, la question qui se pose maintenant est celle de l'outil à utiliser. En fait, la supervision peut se faire de plusieurs manières et il y a toute une diversité d'outils, à surligner : Nagios, Zabbix, Centreon. Mais il existe également d'autres solutions de supervision qui sont moins utilisés, parmi eux: Shinken, Cacti, Eon.

Zabbix, Centreon et Nagios sont considérés comme les leaders du marché. D'ailleurs, on retrouve Zabbix particulièrement dans les banques, et Centreon dans les grandes institutions. Mais le vrai patron et le meilleur outil est celui de Nagios.

La raison principale pour laquelle on opte souvent pour NAGIOS, c'est le nombre de fonctionnalités offertes par le logiciel gratuitement. Il sert à surveiller les systèmes, les réseaux et l'infrastructure. Il offre des services d'alertes en cas

d'incidents, d'erreurs ou d'anomalies. Et même un deuxième avertissement si le problème n'est pas résolu.

Encore plus loin, NAGIOS est un logiciel open source. L'utilisateur est permis à personnaliser et étendre l'application jusqu'à ce qu'elle réponde à ses besoins. Et en fait, plus de 1000 plugins ont déjà été développés en ce sens.

NAGIOS est disponible sous deux versions principales.

- **NAGIOS core** : est la version gratuite et open source de ce logiciel, elle nécessite des connaissances dites relativement profondes dans le domaine de l'administration. Dans cette version, on s'appuie sur le command line.
- **NAGIOS XI** : quant à elle, est la version payante du logiciel, elle est au contraire faite pour des utilisateurs non techniques. Elle comporte beaucoup plus de fonctionnalités qui facilitent la mise à jour des données du super user, ainsi pour donner une image complète, simplifiée et détaillée sur l'état du système.

Description du travail - objectifs :

Dans les pages à suivre, on va procéder à l'installation et la configuration de l'application NAGIOS core dans sa dernière version 4.4.6 sous le système Linux – ubuntu, ainsi que de faire un exemple pour tester le fonctionnement de l'application.

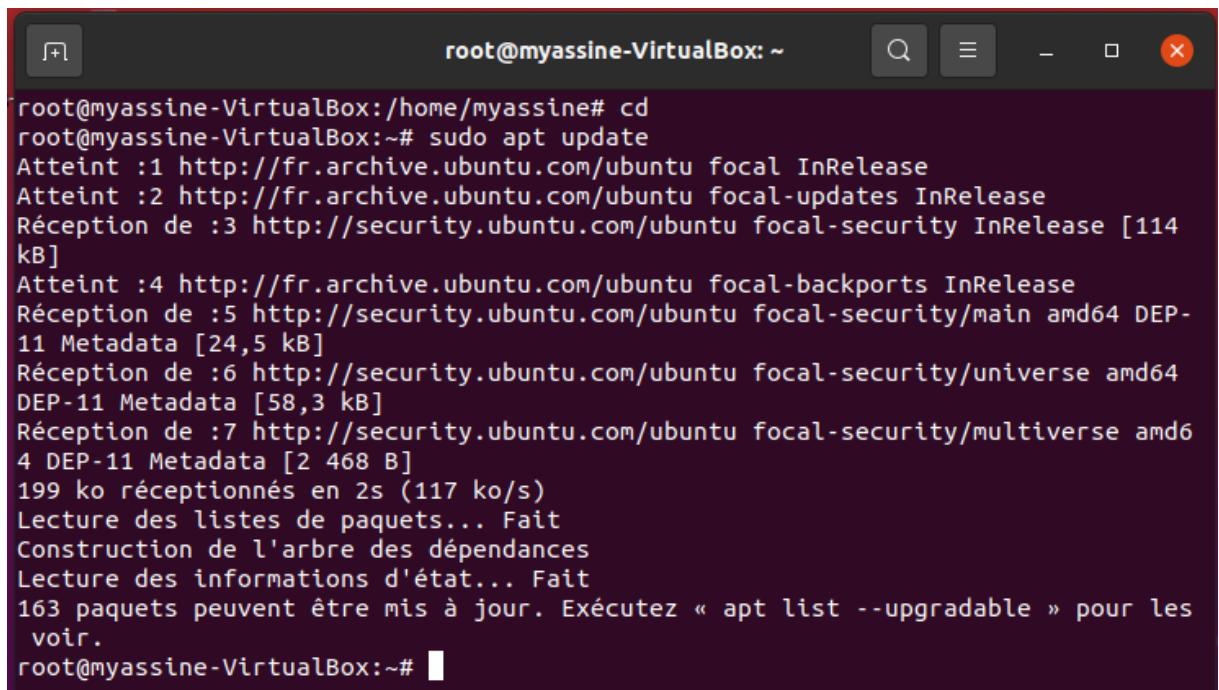
Table du contenu

- I. Introduction**
- II. Description NAGIOS**
- III. Table du contenu**
- IV. Installation NAGIOS**
- V. Configuration NAGIOS**
- VI. Tester le fonctionnement de NAGIOS**
- VII. Dictionnaire des commandes linux utilisés pendant le processus**

Etape 1 : La mise à jour du système

1. Pour assurer la compatibilité entre les éléments à installer et notre système, on commence par la mise à jour de notre système, pour ce faire, on utilise les commandes suivantes :

sudo apt update && sudo apt upgrade -y



```
root@myassine-VirtualBox:~# cd
root@myassine-VirtualBox:~# sudo apt update
Atteint :1 http://fr.archive.ubuntu.com/ubuntu focal InRelease
Atteint :2 http://fr.archive.ubuntu.com/ubuntu focal-updates InRelease
Réception de :3 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Atteint :4 http://fr.archive.ubuntu.com/ubuntu focal-backports InRelease
Réception de :5 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [24,5 kB]
Réception de :6 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [58,3 kB]
Réception de :7 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 DEP-11 Metadata [2 468 B]
199 ko réceptionnés en 2s (117 ko/s)
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
163 paquets peuvent être mis à jour. Exécutez « apt list --upgradable » pour les voir.
root@myassine-VirtualBox:~#
```

```
root@myassine-VirtualBox:~# apt-get upgrade -y
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
Les paquets suivants ont été conservés :
  ubuntu Advantage-tools
Les paquets suivants seront mis à jour :
  alsu-ucm-conf alsu-utils apt apt-utils bluez bluez-cups bluez-obexd dirmngr
  evince evince-common evolution-data-server evolution-data-server-common
  firefox firefox-locale-en firefox-locale-fr fonts-noto-color-emoji
  fonts-opensymbol fprintd friendly-recovery gir1.2-goa-1.0 gir1.2-mutter-6
  gir1.2-secret-1 gjs gnome-control-center gnome-control-center-data
  gnome-control-center-faces gnome-disk-utility gnome-online-accounts
  gnome-shell gnome-shell-common gnome-shell-extension-appindicator gnupg
  gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client gpg-wks-server gpgconf
  gpgsm gpgv grub-common grub-pc grub-pc-bin grub2-common iio-sensor-proxy
  initramfs-tools initramfs-tools-bin initramfs-tools-core libapt-pkg6.0
  libasound2 libasound2-data libatopology2 libbluetooth3 libcamel-1.2-62
  libebackend-1.2-10 libebook-1.2-20 libebook-contacts-1.2-3 libecal-2.0-1
  libedata-book-1.2-26 libedata-cal-2.0-1 libedataserver-1.2-24
  libedataserverui-1.2-2 libevdocument3-4 libevview3-3 libgjs0g libgoa-1.0-0b
  libgoa-1.0-common libgoa-backend-1.0-1 libhogweed5 libjuh-java libjurt-java
  libldap-2.4-2 libldap-common libmutter-6-0 libnetplan0 libnettle7
  libnss-systemd libpam-fprintd libpam-modules libpam-modules-bin
  libpam-runtime libpam-systemd libpam0g libpcie3 libprocps8
  libpulse-mainloop-glib0 libpulse0 libpulsedsp libreoffice-base-core
  libreoffice-calc libreoffice-common libreoffice-core libreoffice-draw
  libreoffice-gnome libreoffice-gtk3 libreoffice-help-common
  libreoffice-help-en-us libreoffice-help-fr libreoffice-impress
  libreoffice-l10n-fr libreoffice-math libreoffice-ogltrans
  libreoffice-pdfimport libreoffice-style-breeze libreoffice-style-colibre
```

Etape 2 : Installation des dépendances de NAGIOS

1. Les dépendances (dependencies) sont des paquets qui ne sont pas visibles à l'utilisateur et qui sont installés lorsqu'il y a des paquets qui dépendent d'autres. Dans notre cas, les dépendances qu'on devra installer sont des paquets dont dépend NAGIOS dans son fonctionnement et qui lui sont recommandés.

Il s'avère donc qu'avant d'installer NAGIOS, on doit installer ses dépendances. Dans ce sens, on installe les dépendances recommandées pour NAGIOS et ce avec la commande suivante : (Tout les éléments de cette commandes sont expliqués dans le dictionnaire des commandes vers la fin du document)

```
sudo apt install -y autoconf bc gawk dc build-essential gcc  
libc6 make wget unzip apache2 php libapache2-mod-php  
libgd-dev libmcrypt-dev make libssl-dev snmp libnet-snmp-  
perl gettext
```

```
myassine@myassine-VirtualBox:~$ sudo apt install -y autoconf bc gawk dc build-es  
sential gcc libc6 make wget unzip apache2 php libapache2-mod-php libgd-dev libmc  
rypt-dev make libssl-dev snmp libnet-snmp-perl gettext  
Lecture des listes de paquets... Fait  
Construction de l'arbre des dépendances  
Lecture des informations d'état... Fait  
bc est déjà la version la plus récente (1.07.1-2build1).  
bc passé en « installé manuellement ».  
dc est déjà la version la plus récente (1.07.1-2build1).  
dc passé en « installé manuellement ».  
unzip est déjà la version la plus récente (6.0-25ubuntu1).  
unzip passé en « installé manuellement ».  
wget est déjà la version la plus récente (1.20.3-1ubuntu1).  
wget passé en « installé manuellement ».  
apache2 est déjà la version la plus récente (2.4.41-4ubuntu3.3).  
libc6 est déjà la version la plus récente (2.31-0ubuntu9.2).  
libc6 passé en « installé manuellement ».
```

```
Setting up libxpm-dev:amd64 (1:3.5.12-1) ...  
Setting up libgd-dev:amd64 (2.2.5-5.2ubuntu2) ...  
Processing triggers for php7.4-cli (7.4.3-4ubuntu2.4) ...  
Processing triggers for libapache2-mod-php7.4 (7.4.3-4ubuntu2.4) ...  
soukaina@soukaina-VirtualBox:~$
```

Etape 3: Téléchargement et extraction de NAGIOS

Après l'installation des dépendances, on devra maintenant installer et extraire NAGIOS.

1. On va tout d'abord créer un répertoire où on va effectuer l'installation, comme suit :
mkdir nagios
et on accède à ce répertoire :
cd nagios

```
soukaina@soukaina-VirtualBox:~$ mkdir nagios  
soukaina@soukaina-VirtualBox:~$ cd nagios  
soukaina@soukaina-VirtualBox:~/nagios$
```

2. Pour télécharger le fichier compressé contenant le fichier exécutable Nagios dans sa version 4.4.6, on utilise la commande :

sudo wget

<https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz>

```
soukaina@soukaina-VirtualBox:~/nagios$ sudo wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
--2021-06-24 18:02:47-- https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11333414 (11M) [application/x-gzip]
Saving to: 'nagios-4.4.6.tar.gz'

nagios-4.4.6.tar.gz      100%[=====] 10,81M  104KB/s    in 3m 5s

2021-06-24 18:05:53 (59,8 KB/s) - 'nagios-4.4.6.tar.gz' saved [11333414/11333414]

soukaina@soukaina-VirtualBox:~/nagios$
```

On vérifie le téléchargement, on utilise la commande ***ll*** pour afficher le contenu du répertoire :

```
soukaina@soukaina-VirtualBox:~/nagios$ ll
total 11080
drwxrwxr-x  2 soukaina soukaina    4096 jun 24 18:02 .
drwxr-xr-x 16 soukaina soukaina    4096 jun 24 17:56 /
-rw-r--r--  1 root     root     11333414 mai  5  2020 nagios-4.4.6.tar.gz
```

Le téléchargement est donc fait correctement.

3. Il nous faut maintenant extraire les fichiers téléchargés, pour ce faire, on utilise la commande suivante :

sudo tar -xvf nagios-4.4.6.tar.gz

Avec : ($-xvf \Leftrightarrow x(extract)$ → extraire, v (verbose) → créer explicitement les fichiers extraits, $f(file)$ → le fichier à suivre est celui à extraire)

```
soukaina@soukaina-VirtualBox:~/nagios$ tar -xvf nagios-4.4.6.tar.gz
nagios-4.4.6/
nagios-4.4.6/.gitignore
nagios-4.4.6/.travis.yml
nagios-4.4.6/CONTRIBUTING.md
nagios-4.4.6/Changelog
nagios-4.4.6/INSTALLING
nagios-4.4.6/LEGAL
nagios-4.4.6/LICENSE
```

```
nagios-4.4.6/xdata/xsddefault.c
nagios-4.4.6/xdata/xsddefault.h
soukaina@soukaina-VirtualBox:~/nagios$ ll
total 11084
drwxrwxr-x  3 soukaina soukaina    4096 jun 24 18:51 .
drwxr-xr-x 16 soukaina soukaina    4096 jun 24 17:56 ../
drwxrwxr-x 20 soukaina soukaina    4096 abr 28 2020 nagios-4.4.6/
-rw-r--r--  1 root     root      113333414 mai  5 2020 nagios-4.4.6.tar.gz
soukaina@soukaina-VirtualBox:~/nagios$
```

L'extraction est achevée avec succès.

```
soukaina@soukaina-VirtualBox:~/nagios$ cd nagios-4.4.6/
soukaina@soukaina-VirtualBox:~/nagios/nagios-4.4.6$ ll
total 616
drwxrwxr-x 20 soukaina soukaina    4096 abr 28 2020 .
drwxrwxr-x  3 soukaina soukaina    4096 jun 24 18:51 ../
-rwxrwxr-x  1 soukaina soukaina     346 abr 28 2020 acllocal.m4*
drwxrwxr-x  2 soukaina soukaina    4096 abr 28 2020 autoconf-macros/
drwxrwxr-x  2 soukaina soukaina    4096 abr 28 2020 base/
drwxrwxr-x  2 soukaina soukaina    4096 abr 28 2020 cgi/
-rw-rw-r--  1 soukaina soukaina   32590 abr 28 2020 Changelog
drwxrwxr-x  2 soukaina soukaina    4096 abr 28 2020 common/
-rwxrwxr-x  1 soukaina soukaina   43765 abr 28 2020 config.guess*
-rwxrwxr-x  1 soukaina soukaina   36345 abr 28 2020 config.sub*
-rwxrwxr-x  1 soukaina soukaina  246354 abr 28 2020 configure*
-rw-rw-r--  1 soukaina soukaina  29812 abr 28 2020 configure.ac
drwxrwxr-x  5 soukaina soukaina    4096 abr 28 2020 contrib/
-rw-rw-r--  1 soukaina soukaina   6291 abr 28 2020 CONTRIBUTING.md
drwxrwxr-x  2 soukaina soukaina    4096 abr 28 2020 docs/
-rw-rw-r--  1 soukaina soukaina    886 abr 28 2020 doxy.conf
```

Etape 4: Installation et exécution NAGIOS :

1. Tout en restant dans le répertoire extrait précédemment, on vérifie si on a tout le nécessaire pour installer NAGIOS (**./configure**). Et puisque NAGIOS a besoin d'un serveur apache pour présenter les pages web NAGIOS, c'est une exigence aussi qui doit être réalisée avant l'installation, on devra donc configurer le host virtuel apache, pour ce faire, on utilise la commande suivante :

sudo ./configure --with-httpd-conf=/etc/apache2/sites-enabled

```
soukaina@soukaina-VirtualBox:~/nagios/nagios-4.4.6$ sudo ./configure --with--httpd-conf=/etc/apache2/sites-enabled
[sudo] password for soukaina:
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
```

La vérification des exigences :

```
checking for ctype.h... yes
checking dirent.h usability... yes
checking dirent.h presence... yes
checking for dirent.h... yes
checking errno.h usability... yes
checking errno.h presence... yes
checking for errno.h... yes
checking fcntl.h usability... yes
checking fcntl.h presence... yes
checking for fcntl.h... yes
checking getopt.h usability... yes
```

Après vérification, si tous les éléments requis sont présents, on nous suggère d'utiliser la commande ***make all*** pour à la fois compiler le programme principal, ainsi que ses CGIs. Les CGIs (pour Common Gateway Interface), sont des scripts qui jouent le rôle d'une interface pour récupérer les pages web demandées d'un serveur web.

```
** Preprocessor (used by main) **

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

soukaina@soukaina-VirtualBox:~/nagios/nagios-4.4.6$
```

On exécute la commande ***make all***

```
soukaina@soukaina-VirtualBox:~/nagios/nagios-4.4.6$ sudo make all
[sudo] password for soukaina:
cd ./base && make
make[1]: Entering directory '/home/soukaina/nagios/nagios-4.4.6/base'
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nagios.o nagios.c
nagios.c: In function 'main':
nagios.c:611:4: warning: ignoring return value of 'asprintf', declared with attribute warn_unused_result [-Wunus
ed-result]
  611 |     asprintf(&mac->x[MACRO_PROCESSSTARTTIME], "%llu", (unsigned long long)program_start);
```

Opération achevée avec succès :

```
*****
Enjoy.

soukaina@soukaina-VirtualBox:~/nagios/nagios-4.4.6$
```

2. L'autre étape qu'on va faire, c'est de créer un groupe utilisateur pour NAGIOS avec la commande : **sudo install-groups-users**

```
soukaina@soukaina-VirtualBox:~/nagios/nagios-4.4.6$ sudo make install-groups-users
[sudo] password for soukaina:
groupadd -r nagios
useradd -g nagios nagios
soukaina@soukaina-VirtualBox:~/nagios/nagios-4.4.6$ █
```

Maintenant, on ajoute l'utilisateur www-data d'apache au groupe NAGIOS, www-data est l'utilisateur utilisé par les serveurs web par défaut pour effectuer des opérations ordinaires, on l'ajoute comme suit :
sudo usermod -a -G nagios www-data

```
soukaina@soukaina-VirtualBox:~/nagios/nagios-4.4.6$ sudo usermod -a -G nagios www-data
soukaina@soukaina-VirtualBox:~/nagios/nagios-4.4.6$ getent group nagios
nagios:x:998:www-data
soukaina@soukaina-VirtualBox:~/nagios/nagios-4.4.6$ █
```

L'utilisateur **www-data** est ajouté avec succès.

3. C'est le moment pour installer les binaries dont on aura besoin puisque NAGIOS dépend dans son fonctionnement sur l'exécution et l'installation de plusieurs programmes. Si on procède par la méthode ordinaire, on risque de perdre du temps et fournir plus d'effort que ce dont on a besoin. Pour éviter ceci, on télécharge les binaires qui n'installent rien, mais qui contient des fichiers qui sont précompilés et directement exécutables.

Pour installer les binaires, on utilise la commande suivante :

Sudo make install

```
lamyae@lomyae-Asr:~/nagios/nagios-4.4.6$ sudo make install
cd ./base && make install
make[1]: Entering directory '/home/lamyae/nagios/nagios-4.4.6/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagiostats /usr/local/nagios/bin
make[1]: Leaving directory '/home/lamyae/nagios/nagios-4.4.6/base'
cd ./cgi && make install
make[1]: Entering directory '/home/lamyae/nagios/nagios-4.4.6/cgi'
make install-basic
make[2]: Entering directory '/home/lamyae/nagios/nagios-4.4.6/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
for file in *.cgi; do \
    /usr/bin/install -c -s -m 775 -o nagios -g nagios $file /usr/local/nagios/sbin; \
done
make[2]: Leaving directory '/home/lamyae/nagios/nagios-4.4.6/cgi'
make[1]: Leaving directory '/home/lamyae/nagios/nagios-4.4.6/cgi'
cd ./html && make install
make[1]: Entering directory '/home/lamyae/nagios/nagios-4.4.6/html'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/media
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/stylesheets
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/contexthelp
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/docs
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/docs/images
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/js
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/images
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/images/logos
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/includes
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/ssi
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/angularjs
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/angularjs/angular-1
.3.9
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/angularjs/ui-utils-
0.2.3
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/bootstrap-3.3.7
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/bootstrap-3.3.7/css
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/d3
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/share/spin
/usr/bin/install -c -m 664 -o nagios -g nagios robots.txt /usr/local/nagios/share
rm -f /usr/local/nagios/share/index.html
rm -f /usr/local/nagios/share/main.html
rm -f /usr/local/nagios/share/side.html
rm -f /usr/local/nagios/share/map.html
rm -f /usr/local/nagios/share/rss-*
rm -rf /usr/local/nagios/share/include/rss
/usr/bin/install -c -m 664 -o nagios -g nagios jsonquery.html /usr/local/nagios/share
/usr/bin/install -c -m 664 -o nagios -g nagios graph-header.html /usr/local/nagios/share
```

Après installation, on nous recommande certaines autres installations :

```
make install-init
  - This installs the init script in /lib/systemd/system

make install-commandmode
  - This installs and configures permissions on the
    directory for holding the external command file

make install-config
  - This installs sample config files in /usr/local/nagios/etc

make[1]: Leaving directory '/home/soukaina/nagios/nagios-4.4.6'
soukaina@soukaina-VirtualBox:~/nagios/nagios-4.4.6$
```

a. Installation du script init :

Les scripts init sont des scripts exécutés pour démarrer des processus dans l'arrière-plan et même dans la frontière graphique, et qui dépendent surtout de la réaction avec l'utilisateur.

Avec la commande : ***sudo make install-init*** on installe ces scripts dans **/usr/local/nagios/etc**

```
lamyae@lmyae-Asr:~/nagios/nagios-4.4.6$ sudo make install-init
/usr/bin/install -c -m 755 -o root -g root /lib/systemd/system
/usr/bin/install -c -m 755 -o root -g root startup/default-service /lib/systemd/system/nagios
.service
lamyae@lmyae-Asr:~/nagios/nagios-4.4.6$
```

b. Installation commandmode :

Cette étape rentre dans la configuration du serveur web, on le prépare pour exécuter des ordres du command line, on installe lui installe les permissions nécessaires.

```
lamyae@lmyae-Asr:~/nagios/nagios-4.4.6$ sudo make install-commandmode
/usr/bin/install -c -m 775 -o nagios -g nagcmd -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw
*** External command directory configured ***
lamyae@lmyae-Asr:~/nagios/nagios-4.4.6$
```

c. Installation des fichiers de configuration :

Les fichiers de configuration sont des fichiers requis pour échanger des données avec la commande line, ils contiennent des informations sur l'expéditeur et le destinataire.

```
lanyae@lanyae-Asr:~/nagios/nagios-4.4.6$ sudo make install-config
make: Warning: File 'Makefile' has modification time 1504 s in the future
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cfg /usr/local/nagios/etc/cgi.cfg
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource.cfg /usr/local/nagios/etc/resource.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/templates.cfg
/usr/local/nagios/etc/objects/templates.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/commands.cfg
/usr/local/nagios/etc/objects/commands.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/contacts.cfg
/usr/local/nagios/etc/objects/contacts.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/timeperiods.cfg
/usr/local/nagios/etc/objects/timeperiods.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/localhost.cfg
/usr/local/nagios/etc/objects/localhost.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/windows.cfg
/usr/local/nagios/etc/objects/windows.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/printer.cfg
/usr/local/nagios/etc/objects/printer.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/switch.cfg
/usr/local/nagios/etc/objects/switch.cfg

*** Config files installed ***

Remember, these are *SAMPLE* config files. You'll need to read
the documentation for more information on how to actually define
services, hosts, etc. to fit your particular needs.

make: warning: Clock skew detected. Your build may be incomplete.
lanyae@lanyae-Asr:~/nagios/nagios-4.4.6$
```

Continuons la configuration du serveur web, on installe les documents de configuration pour apache :
sudo make install-webconf

```
lanyae@lanyae-Asr:~/nagios/nagios-4.4.6$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/sites-enabled/nagios.conf
if [ 0 -eq 1 ]; then \
    ln -s /etc/apache2/sites-enabled/nagios.conf /etc/apache2/sites-enabled/nagios.conf;
\
fi

*** Nagios/Apache conf file installed ***
lanyae@lanyae-Asr:~/nagios/nagios-4.4.6$
```

Maintenant, on active les deux modules **rewrite** et **cgi** pour **apache**, pour ce faire, on utilise **a2enmod** pour transférer des modules à partir de **/etc/apache2/mods-available** le fichier des modules disponibles vers

/etc/apache2/mods-enabled le fichier des modules activés.

Le module **rewrite** donne des permissions pour des manipulations des URLs hors le serveur pour être réécrits selon des règles.

Le module **cgi** quant à lui, (common gateway interface), donne au serveur la capacité d'exécuter des programmes externes, notamment pour traiter les demandes des clients.

Pour activer les deux modules, on utilise la commande :
sudo a2enmod rewrite cgi

```
soukaina@soukaina-VirtualBox:~/nagios/nagios-4.4.6$ sudo a2enmod rewrite cgi
[sudo] password for soukaina:
Enabling module rewrite.
Enabling module cgi.
To activate the new configuration, you need to run:
  systemctl restart apache2
soukaina@soukaina-VirtualBox:~/nagios/nagios-4.4.6$ █
```

Pour pouvoir accéder à l'interface d'administration web, on devra créer des données d'authentification pour notre administrateur NAGIOS, pour ce faire, on utilise la commande **htpasswd** qui sert à mettre à jour le fichier contenant les données d'authentification

/usr/local/nagios/etc/htpasswd.users, par nos données en créant un nouvel utilisateur nagiosadmin pour lequel on choisit un mot de passe.

On exécute :

sudo htpasswd -c

/usr/local/nagios/etc/htpasswd.users nagiosadmin

```
|fatima@fatima-VirtualBox:~/nagios/nagios-4.4.6$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
fatima@fatima-VirtualBox:~/nagios/nagios-4.4.6$ |
```

Etape 5: Installation et configuration des plugins NAGIOS et de NRPE

Certes, pour garantir un fonctionnement optimal, NAGIOS est censé détecter les erreurs et renvoyer des alertes ou des notifications à l'administrateur. Cette fonctionnalité est fournie grâce aux plugins.

Les plugins sont des programmes compilés ou des scripts pouvant être exécutés à l'aide du terminal. Ils servent à contrôler l'état d'une hôte, une machine ou un service dans le système.

Pour surligner l'importance des plugins, on visite l'interface web. On procède comme suit :

On redémarre et on active l'apache et NAGIOS pour que l'interface web fonctionne :

```
soukaina@soukaina-VirtualBox:~$ sudo systemctl restart apache2
soukaina@soukaina-VirtualBox:~$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
soukaina@soukaina-VirtualBox:~$ sudo systemctl restart nagios
soukaina@soukaina-VirtualBox:~$ sudo systemctl enable nagios
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /lib/systemd/system/nagios.service
```

```
soukaina@soukaina-VirtualBox:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2021-06-25 18:15:59 WEST; 1min 42s ago
     Docs: https://httpd.apache.org/docs/2.4/
 Main PID: 57560 (apache2)
    Tasks: 6 (limit: 3888)
   Memory: 9.9M
      CGrou... /system.slice/apache2.service
              └─57560 /usr/sbin/apache2 -k start

...
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2021-06-25 18:15:59 WEST; 1min 42s ago
     Docs: https://httpd.apache.org/docs/2.4/
 Main PID: 57560 (apache2)
    Tasks: 6 (limit: 3888)
   Memory: 9.9M
      CGrou... /system.slice/apache2.service
              ├─57560 /usr/sbin/apache2 -k start
              ├─57561 /usr/sbin/apache2 -k start
              ├─57562 /usr/sbin/apache2 -k start
```

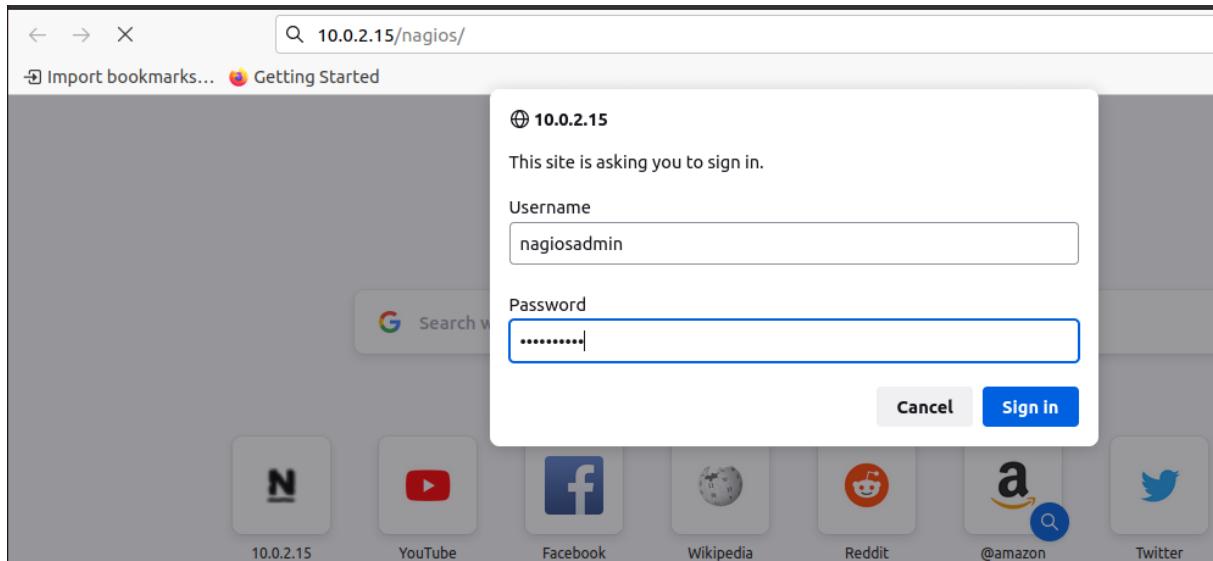
On récupère notre adresse IP :

```
soukaina@soukaina-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
          inet6 fe80::283:4a8a:4158:c33d prefixlen 64 scopeid 0x20<link>
            ether 08:00:27:78:73:37 txqueuelen 1000 (Ethernet)
              RX packets 348304 bytes 489847424 (489.8 MB)
              RX errors 0 dropped 0 overruns 0 frame 0
              TX packets 114074 bytes 8691711 (8.6 MB)
              TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
          inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
              RX packets 2451 bytes 359036 (359.0 KB)
              RX errors 0 dropped 0 overruns 0 frame 0
              TX packets 2451 bytes 359036 (359.0 KB)
              TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

soukaina@soukaina-VirtualBox:~$
```

On l'utilise pour intégrer l'interface et on s'identifie par les données créées précédemment :



On ouvre la rubrique **Alerts** :

```

[06-25-2021 18:26:04] HOST ALERT: localhost;DOWN;HARD;10;(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_ping,...) failed. errno is 2: No such file or director
[06-25-2021 18:25:04] HOST ALERT: localhost;DOWN;SOFT;9;(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_ping,...) failed. errno is 2: No such file or director
[06-25-2021 18:24:04] HOST ALERT: localhost;DOWN;SOFT;8;(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_ping,...) failed. errno is 2: No such file or director
[06-25-2021 18:23:04] HOST ALERT: localhost;DOWN;SOFT;7;(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_ping,...) failed. errno is 2: No such file or director
[06-25-2021 18:22:04] HOST ALERT: localhost;DOWN;SOFT;6;(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_ping,...) failed. errno is 2: No such file or director
[06-25-2021 18:22:04] SERVICE ALERT: localhost;Total Processes;CRITICAL;HARD;1;(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_procs,...) failed. errno is 2: No such file or director
[06-25-2021 18:21:26] SERVICE ALERT: localhost;Swap Usage;CRITICAL;HARD;1;(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_swap,...) failed. errno is 2: No such file or director
[06-25-2021 18:21:04] HOST ALERT: localhost;DOWN;SOFT;5;(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_ping,...) failed. errno is 2: No such file or director
[06-25-2021 18:20:49] SERVICE ALERT: localhost;SSH;CRITICAL;HARD;1;(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_ssh,...) failed. errno is 2: No such file or director
[06-25-2021 18:20:11] SERVICE ALERT: localhost;Root Partition;CRITICAL;HARD;1;(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_disk,...) failed. errno is 2: No such file or director
[06-25-2021 18:20:04] HOST ALERT: localhost;DOWN;SOFT;4;(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_ping,...) failed. errno is 2: No such file or director
[06-25-2021 18:19:34] SERVICE ALERT: localhost;PING;CRITICAL;HARD;1;(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_ping,...) failed. errno is 2: No such file or director
[06-25-2021 18:19:04] HOST ALERT: localhost;DOWN;SOFT;3;(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/check_ping,...) failed. errno is 2: No such file or director

```

Et on observe que NAGIOS relève déjà des erreurs qui nous informent que NAGIOS cherche certains fichiers qui n'existent pas. Ces fichiers représentent les plugins.

Dans la suite, on va procéder pour installer les plugins nécessaires pour le fonctionnement de NAGIOS.

On devra installer deux types de plugins :

- **Monitoring plugins** : qui nous permettent le suivi des hôtes, des services et des programmes et l'évaluation de leurs états.
- **NRPE plugin** : Nagios Remote Plugin Executor plugin, il permet de surveiller le système à distance à l'aide des scripts qui sont sur les systèmes distants. Il permet de surveiller les ressources et les équipements.

```
fatima@fatima-VirtualBox:~/nagios/nagios-4.4.6$ cd ..
fatima@fatima-VirtualBox:~/nagios$ sudo apt install monitoring-plugins nagios-nrpe-plugin -y
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
 libdbi1 libpq5 libradcli4 libtirpc-common libtirpc3 monitoring-plugins-basic monitoring-plugins-common
 monitoring-plugins-standard python3-crypto python3-gpg python3-ldb python3-samba python3-tdb rpcbind
 samba-common samba-common-bin samba-dsdb-modules smbclient
Paquets suggérés :
 icinga | icinga2 nagios-plugins-contrib fping postfix | sendmail-bin | exim4-daemon-heavy
 | exim4-daemon-light qstat heimdal-clients python3-markdown python3-dnspython cifs-utils
Les NOUVEAUX paquets suivants seront installés :
 libdbi1 libpq5 libradcli4 libtirpc-common libtirpc3 monitoring-plugins monitoring-plugins-basic
 monitoring-plugins-common monitoring-plugins-standard nagios-nrpe-plugin python3-crypto python3-gpg
 python3-ldb python3-samba python3-tdb rpcbind samba-common samba-common-bin samba-dsdb-modules smbclient
0 mis à jour, 20 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 4,897 ko dans les archives.
Après cette opération, 32.2 Mo d'espace disque supplémentaires seront utilisés.
Réception de :1 http://ma.archive.ubuntu.com/ubuntu focal/main amd64 libtirpc-common all 1.2.5-1 [7,632 B]
Réception de :2 http://ma.archive.ubuntu.com/ubuntu focal/main amd64 libtirpc3 amd64 1.2.5-1 [77.2 kB]
Réception de :3 http://ma.archive.ubuntu.com/ubuntu focal/main amd64 rpcbind amd64 1.2.5-8 [42.8 kB]
Réception de :4 http://ma.archive.ubuntu.com/ubuntu focal-updates/main amd64 samba-common all 2:4.11.6+dfsg-0ubuntu1.9 [68.7 kB]
Réception de :5 http://ma.archive.ubuntu.com/ubuntu focal-updates/main amd64 smbclient amd64 2:4.11.6+dfsg-0ubuntu1.9 [365 kB]
Réception de :6 http://ma.archive.ubuntu.com/ubuntu focal/main amd64 libdbi1 amd64 0.9.0-5 [27.3 kB]
Réception de :7 http://ma.archive.ubuntu.com/ubuntu focal-updates/main amd64 libpq5 amd64 12.7-0ubuntu0.20.04.1 [
```

```
Creating config file /etc/nagios-plugins/config/mysql.cfg with new version
Creating config file /etc/nagios-plugins/config/netware.cfg with new version
Creating config file /etc/nagios-plugins/config/nt.cfg with new version
Creating config file /etc/nagios-plugins/config/pgsql.cfg with new version
Creating config file /etc/nagios-plugins/config/radius.cfg with new version
Creating config file /etc/nagios-plugins/config/rpc-nfs.cfg with new version
Creating config file /etc/nagios-plugins/config/snmp.cfg with new version
Paramétrage de monitoring-plugins (2.2-6ubuntu1.1) ...
Paramétrage de samba-common-bin (2:4.11.6+dfsg-0ubuntu1.9) ...
Checking smb.conf with testparm
Load smb config files from /etc/samba/smb.conf
Loaded services file OK.
WARNING: The 'netbios name' is too long (max. 15 chars).

Server role: ROLE_STANDALONE

Done
Traitement des actions différées (« triggers ») pour systemd (245.4-4ubuntu3.7) ...
Traitement des actions différées (« triggers ») pour man-db (2.9.1-1) ...
Traitement des actions différées (« triggers ») pour libc-bin (2.31-0ubuntu9.2) ...
fatima@fatima-VirtualBox:~/nagios$
```

Dans cette étape, on se met dans le répertoire **/usr/local/nagios** et on affiche son contenu. C'est le répertoire natif de NAGIOS.

```
hafsa@hafsa-VirtualBox:~/nagios$ cd /usr/local/nagios/etc/
hafsa@hafsa-VirtualBox:/usr/local/nagios/etc$ ll
total 84
drwxrwxr-x 3 nagios nagios 4096 Jun 21 00:57 .
drwxr-xr-x 8 root  root  4096 Jun 21 00:28 ..
-rw-rw-r-- 1 nagios nagios 13710 Jun 21 00:28 cgi.cfg
-rw-r--r-- 1 root  root   50 Jun 21 00:57 htpasswd.users
-rw-rw-r-- 1 nagios nagios 45843 Jun 21 00:28 nagios.cfg
drwxrwxr-x 2 nagios nagios 4096 Jun 21 00:28 objects/
-rw-rw---- 1 nagios nagios 1312 Jun 21 00:28 resource.cfg
hafsa@hafsa-VirtualBox:/usr/local/nagios/etc$
```

Ce répertoire contient quatre fichiers principaux et un répertoire :

- Le fichier « **htpasswd.users** » (créé lors de l'installaion de Nagios), permet de protéger l'accès à l'interface web d'administration par un mot de passe htaccess.
- Le fichier « **nagios.cfg** » est le fichier de configuration principal de Nagios. Et c'est la premiere chose que le Nagios va lire lors de démarrage.
- Le fichier « **resource.cfg** » contient des paramètres spécifiques à nos **USER MACROS**.
- Le fichier « **cgi.cfg** » dédié à la configuration de l'interface d'administration (aussi les droits utilisateur).
- Le répertoire « **objects** » contient huit fichiers de configuration avec une extension **.cfg**

Parmi les configurations requises, la configuration des hôtes. Des configurations similaires se font au niveau du fichier **nagios.cfg**.

```
soukaina@soukaina-VirtualBox:/usr/local/nagios/etc$ cat nagios.cfg
#####
#
# NAGIOS.CFG - Sample Main Config File for Nagios 4.4.6
#
# Read the documentation for more information on this configuration
# file. I've provided some comments here, but things may not be so
# clear without further explanation.
#
#
#####

# LOG FILE
# This is the main log file where service and host events are logged
# for historical purposes. This should be the first option specified
# in the config file!!!

log_file=/usr/local/nagios/var/nagios.log
```

Les configurations des hôtes existent dans les lignes 51,52,53,54 en commentaire. Pour les activer, on les décommente, en éditant ce fichier dans l'éditeur de texte.

```
asmaa@asmaa-virtual-machine:/usr/local/nagios/etc$ sudo vim nagios.cfg
```

```
50
51 #cfg_dir=/usr/local/nagios/etc/servers
52 #cfg_dir=/usr/local/nagios/etc/printers
53 #cfg_dir=/usr/local/nagios/etc/switches
54 #cfg_dir=/usr/local/nagios/etc/routers
55
```

```
51 cfg_dir=/usr/local/nagios/etc/servers
52 cfg_dir=/usr/local/nagios/etc/printers
53 cfg_dir=/usr/local/nagios/etc/switches
54 cfg_dir=/usr/local/nagios/etc/routers
55
```

Maintenant, on crée les répertoires de configurations des hôtes physiquement.

```
asmaa@asmaa-virtual-machine:/usr/local/nagios/etc$ sudo mkdir servers printers
switches routers
[sudo] password for asmaa:
asmaa@asmaa-virtual-machine:/usr/local/nagios/etc$ ll
total 100
drwxrwxr-x 7 nagios nagios 4096 22:55 24 يوزبو .
drwxr-xr-x 8 root root 4096 22:11 24 يوزبو ..
-rw-rw-r-- 1 nagios nagios 13710 22:11 24 يوزبو cgi.cfg
-rw-r--r-- 1 root root 50 22:23 24 يوزبو htpasswd.users
-rw-rw-r-- 1 nagios nagios 45839 22:50 24 يوزبو nagios.cfg
drwxrwxr-x 2 nagios nagios 4096 22:11 24 يوزبو objects/
drwxr-xr-x 2 root root 4096 22:55 24 يوزبو printers/
-rw-rw---- 1 nagios nagios 1312 22:11 24 يوزبو resource.cfg
drwxr-xr-x 2 root root 4096 22:55 24 يوزبو routers/
drwxr-xr-x 2 root root 4096 22:55 24 يوزبو servers/
drwxr-xr-x 2 root root 4096 22:55 24 يوزبو switches/
```

Et on vérifie la création des répertoires :

```
fatima@fatima-VirtualBox:/usr/local/nagios/etc$ sudo mkdir servers printers switches routers
fatima@fatima-VirtualBox:/usr/local/nagios/etc$ ll
total 100
drwxrwxr-x 7 nagios nagios 4096 16:59 24 يونيوبو .
drwxr-xr-x 8 root  root  4096 16:19 24 يونيوبو ..
-rw-rw-r-- 1 nagios nagios 13710 16:19 24 يونيوبو cgi.cfg
-rw-r--r-- 1 root  root   50 16:29 24 يونيوبو htpasswd.users
-rw-rw-r-- 1 nagios nagios 45839 16:57 24 يونيوبو nagios.cfg
drwxrwxr-x 2 nagios nagios 4096 16:19 24 يونيوبو objects/
drwxr-xr-x 2 root  root  4096 16:59 24 يونيوبو printers/
-rw-rw---- 1 nagios nagios 1312 16:19 24 يونيوبو resource.cfg
drwxr-xr-x 2 root  root  4096 16:59 24 يونيوبو routers/
drwxr-xr-x 2 root  root  4096 16:59 24 يونيوبو servers/
drwxr-xr-x 2 root  root  4096 16:59 24 يونيوبو switches/
fatima@fatima-VirtualBox:/usr/local/nagios/etc$
```

Le fichier **resource.cfg** quant à lui, comme cité précédemment, contient nos paramètres des **USER MACROS**.

Les **MACROS** sont des instructions qui réalisent des tâches, une seule instruction se développe automatiquement pourra réaliser plusieurs tâches.

Les **USER MACROS** d'autre part, sont utilisés si on souhaite configurer nos propres **MACROS**, pour effectuer des tâches spécifiques.

On affiche le contenu de ce fichier :

```
fatima@fatima-VirtualBox:/usr/local/nagios/etc$ sudo cat resource.cfg
[sudo] Mot de passe de fatima :
#####
#
# RESOURCE.CFG - Sample Resource File for Nagios 4.4.6
#
#
# You can define $USERx$ macros in this file, which can in turn be used
# in command definitions in your host config file(s). $USERx$ macros are
# useful for storing sensitive information such as usernames, passwords,
# etc. They are also handy for specifying the path to plugins and
# event handlers - if you decide to move the plugins or event handlers to
# a different directory in the future, you can just update one or two
# $USERx$ macros, instead of modifying a lot of command definitions.
#
# The CGIs will not attempt to read the contents of resource files, so
# you can set restrictive permissions (600 or 660) on them.
#
# Nagios supports up to 256 $USERx$ macros ($USER1$ through $USER256$)
#
# Resource files may also be used to store configuration directives for
# external data sources like MySQL...
#
#####
#
# Sets $USER1$ to be the path to the plugins
#$USER1$=/usr/local/nagios/libexec
$USER1$=/usr/lib/nagios/plugins
# Sets $USER2$ to be the path to event handlers
#$USER2$=/usr/local/nagios/libexec/eventhandlers
#
# Store some usernames and passwords (hidden from the CGIs)
#$USER3$=someuser
#$USER4$=somepassword
fatima@fatima-VirtualBox:/usr/local/nagios/etc$
```

Comme cité dans la description du fichier, on en aura besoin pour spécifier le chemin des plugins.

Pour ce faire, on crée notre propre **USER MACROS** comme suit :

\$USER1\$=/usr/lib/nagios/plugins

Pour ce faire, on ouvre le fichier dans l'éditeur de texte, le chemin avant spécification est dans la ligne 26 :

```
# Sets $USER1$ to be the path to the plugins
$USER1$=/usr/local/nagios/libexec
```

On commente ce **USER MACROS** et on ajoute le notre :

```
24 # Sets $USER1$ to be the path to the plugins
25 #$USER1$=/usr/local/nagios/libexec
26 $USER1$=/usr/lib/nagios/plugins
27 # Sets $USER2$ to be the path to event handlers
28 #$USER2$=/usr/local/nagios/libexec/eventhandlers
29
```

Maintenant on devra accéder au répertoire objects, il contient huit fichiers, ces 8 fichiers contiennent des modèles de configuration fournis directement par Nagios, nommés **TEMPLATES**. Et qui permettent de factoriser beaucoup de tâches, ce qui permet de gagner plus du temps sur la configuration et la maintenance des sondes.

```
lanyae@lanyae-Asr:~$ ls -l /usr/local/nagios/etc/objects
total 52
-rw-rw-r-- 1 nagios nagcmd 6747 21:43 20 یو:یو commands.cfg
-rw-rw-r-- 1 nagios nagcmd 1797 21:43 20 یو:یو contacts.cfg
-rw-rw-r-- 1 nagios nagcmd 4777 21:43 20 یو:یو localhost.cfg
-rw-rw-r-- 1 nagios nagcmd 3001 21:43 20 یو:یو printer.cfg
-rw-rw-r-- 1 nagios nagcmd 3484 21:43 20 یو:یو switch.cfg
-rw-rw-r-- 1 nagios nagcmd 12533 21:43 20 یو:یو templates.cfg
-rw-rw-r-- 1 nagios nagcmd 3512 21:43 20 یو:یو timeperiods.cfg
-rw-rw-r-- 1 nagios nagcmd 4074 21:43 20 یو:یو windows.cfg
lanyae@lanyae-Asr:~$
```

Parmi ces huit fichiers, **contacts.cfg**. Il contient les informations sur les contacts auxquels on va envoyer les notifications.

```
asmaa@asmaa-virtual-machine:/usr/local/nagios/etc$ cd objects/
asmaa@asmaa-virtual-machine:/usr/local/nagios/etc/objects$ █
```

Il nous faut donc modifier ce fichier et ajouter nos informations pour être notifiés. Ceci se fera dans l'éditeur de texte.

```
asmaa@asmaa-virtual-machine:/usr/local/nagios/etc/objects$ sudo vim contacts.c
fg█
```

```
#####
#  

# CONTACTS.CFG - SAMPLE CONTACT/CONTACTGROUP DEFINITIONS  

#  

#  

# NOTES: This config file provides you with some example contact and contact  

# group definitions that you can reference in host and service  

# definitions.  

#  

# You don't need to keep these definitions in a separate file from your  

# other object definitions. This has been done just to make things  

# easier to understand.  

#  

#####  

#  

#  

# CONTACTS  

#  

@  

"contacts.cfg" 51L, 1797C
```

1,1 Top

```
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             asmaaelfadil32@gmail.com ; <>***** CHANGE THIS TO  

YOUR EMAIL ADDRESS *****  

}
```

Pour configurer le contrôle à distance, on modifie le fichier **commands.cfg** :

```
asmaa@asmaa-virtual-machine:/usr/local/nagios/etc/objects$ sudo vim commands.c  

fg
```

```
#####
#  

# COMMANDS.CFG - SAMPLE COMMAND DEFINITIONS FOR NAGIOS 4.4.6  

#  

#  

# NOTES: This config file provides you with some example command definitions  

# that you can reference in host, service, and contact definitions.  

#  

# You don't need to keep commands in a separate file from your other  

# object definitions. This has been done just to make things easier to  

# understand.  

#  

#####  

##  

#  

# SAMPLE NOTIFICATION COMMANDS  

#  

# These are some example notification commands. They may or may not work on  

# your system without modification. As an example, some systems will require  

"commands.cfg" 253L, 6747C
```

1,1 Top

A la fin de la ligne, on ajoute cette commande à la fin de la ligne :

```
define command {  
    command_name check_nrpe  
    command_line $USER1$/check_nrpe -H  
    $HOSTADDRESS$ -c $ARG1$
```

```
}
```

```
# host_perfdata_file and service_perfdata_file options in the main config file  
. # #####  
##  
##  
define command {  
    command_name process-host-perfdata  
    command_line /usr/bin/printf "%b" "$LASTHOSTCHECK$\t$HOSTNAME$\t$HOSTST  
ATE$\t$HOSTATTEMPT$\t$HOSTSTATETYPE$\t$HOSTEXECUTIONTIME$\t$HOSTOUTPUT$\t$HOST  
PERFDATA$\n" >> /usr/local/nagios/var/host-perfdata.out  
}  
  
define command {  
    command_name process-service-perfdata  
    command_line /usr/bin/printf "%b" "$LASTSERVICECHECK$\t$HOSTNAME$\t$SER  
VICEDESC$\t$SERVICESTATE$\t$SERVICEATTEMPT$\t$SERVICESTATETYPE$\t$SERVICEEXECU  
TIONTIME$\t$SERVICELATENCY$\t$SERVICEOUTPUT$\t$SERVICEPERFDATA$\n" >> /usr/loc  
al/nagios/var/service-perfdata.out  
}
```

```

# These are sample performance data commands that can be used to send performance
|# data output to two text files (one for hosts, another for services). If you
# plan on simply writing performance data out to a file, consider using the
# host_perfdata_file and service_perfdata_file options in the main config file.
#
#####
define command {

    command_name process-host-perfdata
    command_line /usr/bin/printf "%b" "$LASTHOSTCHECK$\t$HOSTNAME$\t$HOSTSTATE$\t$HOSTATTEMPT$\t$HOSTSTATETYPE
\$t$HOSTEXECUTIONTIME$\t$HOSTOUTPUT$\t$HOSTPERFDATA$\n" >> /usr/local/nagios/var/host-perfdata.out
}

define command {
    command_name process-service-perfdata
    command_line /usr/bin/printf "%b" "$LASTSERVICECHECK$\t$HOSTNAME$\t$SERVICEDESC$\t$SERVICESTATE$\t$SERVICE
ATTEMPT$\t$SERVICESTATETYPE$\t$SERVICEEXECUTIONTIME$\t$SERVICELATENCY$\t$SERVICEOUTPUT$\t$SERVICEPERFDATA$\n" >>
/usr/local/nagios/var/service-perfdata.out
}
define command{
    command_name check_nrpe
    command_line $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
}
:x|

```

Ceci est la définition de la commande du service **NRPE** qui sert à exécuter des commandes de supervision sur des machines distantes. Elle se définit dans les balises define command {} avec comme variables :

- **command_name**: nom de la commande, tel que précisé dans la variable check_command de la définition du service (dans notre cas c'est **check_nrpe**)
- **command_line**: commandes de check du service
- **La variable \$USER1\$** pointe sur le dossier **/usr/local/nagios/libexec/** où sont stockés tous les scripts de check de services de Nagios.
- **L'option -H** permet de stipuler l'hôte sur lequel le check du service va se faire, et l'option -c est parmi les options qu'on peut associer à la commande.

Finalement, on se connecte une autre fois dans Nagios pour vérifier si les erreurs qu'on a trouvé précédemment dans la rubrique **Alerts** sont corrigés après l'installation des plugins.

The screenshot shows the Nagios Core 4.4.6 homepage. At the top right, it displays "Version 4.4.6" and the date "April 28, 2020". A "Check for updates" link is also present. The main content area features three large download buttons for "Nagios XI", "Nagios Log Server", and "Nagios Network Analyzer", each accompanied by a small monitor icon. Below these are several sections: "Get Started" (with a list of monitoring steps), "Quick Links" (listing Nagios Library, Labs, Exchange, Support, and official sites), "Latest News" (listing recent Nagios updates), and "Don't Miss..." (listing monitoring log data, Nagios Network Analyzer capabilities, and the availability of Nagios XI 5). The left sidebar contains navigation links for General, Current Status, Service Groups, Problems, Reports, and Notifications.

The screenshot shows the "All Hosts and Services" page of Nagios Core 4.4.6. It includes a "Log File Navigation" section with a "Latest Archive" button, a date range from "Fri Jun 25 00:00:00 +01 2021" to "Present..", and a log file path "File: /usr/local/nagios/var/nagios.log". To the right, there are "State type options" (set to "All state types") and "History detail level for all hosts" (set to "All alerts"). A checkbox group allows hiding flapping, downtime, process messages, or older entries. A red message at the bottom states "No history information was found in the current log file". The left sidebar is identical to the one in the previous screenshot, showing General, Current Status, Service Groups, Problems, Reports, and Notifications.

On voit qu'aucun erreur a été détecté.

Test du fonctionnement Nagios

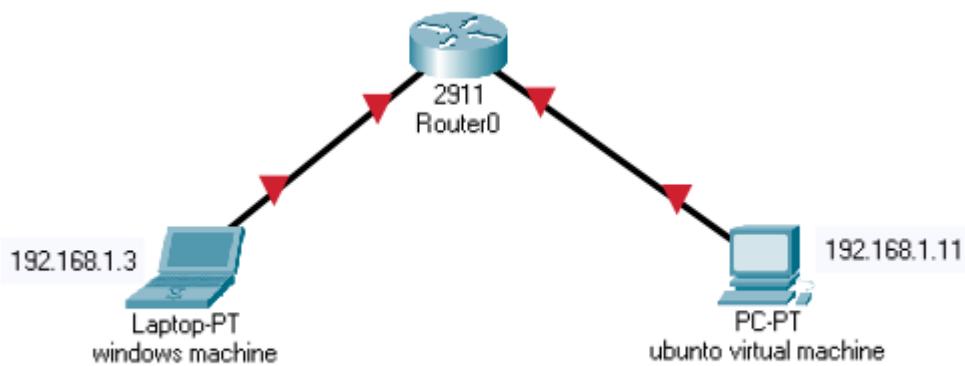
Introduction :

Après l'installation et la configuration du logiciel Nagios, on passera par la suite à le tester sur un exemple réel. On pourra ainsi détecter ses anomalies, les corriger, mais aussi on vérifiera le fonctionnement de notre serveur Nagios. On s'assure ainsi de la bonne configuration de l'application, et d'une supervision meilleure de notre système. Pour ce faire, on va associer un hôte Windows à notre serveur nagios et la configurer proprement pour être supervisée par nagios. Ce qui pourra nous permettre de détecter ainsi corriger les erreurs qui vont apparaître.

Cahier des charges

1- Le schéma de réseau à superviser

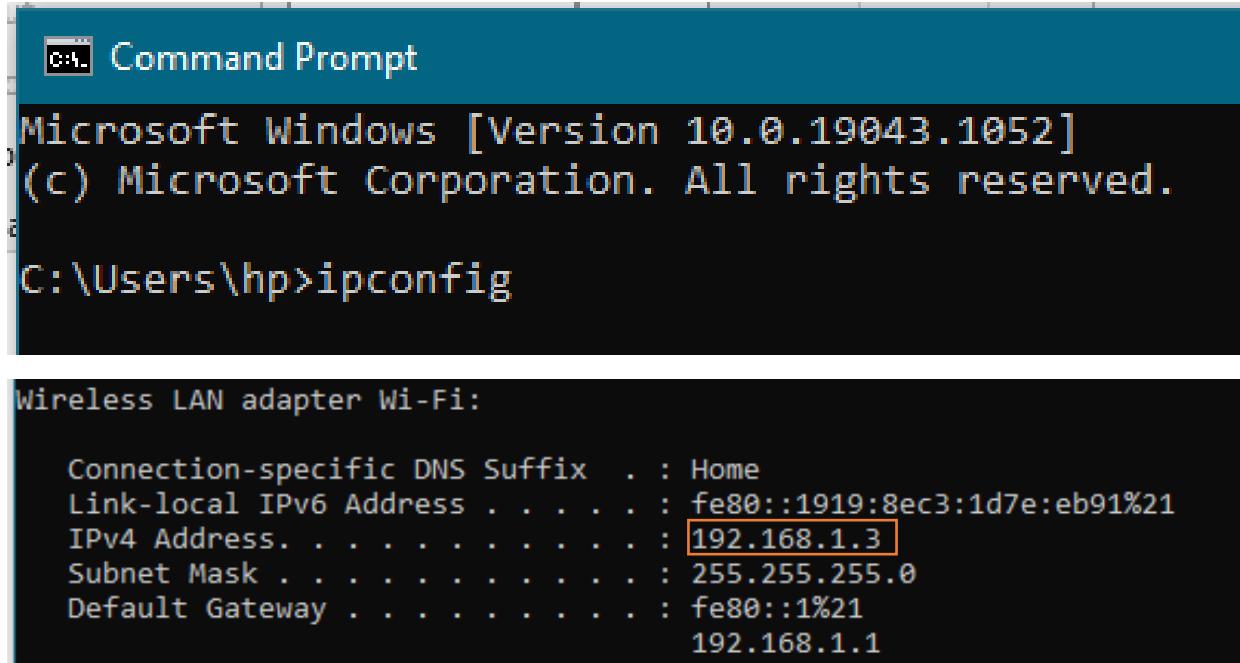
Pour atteindre notre but, on va s'appuyer sur le réseau suivant. Qu'on a réalisé dans le logiciel Packet trace :



On récupère notre IP sur les deux machines hôtes et serveur :

- **Dans la machine Windows :**

On tape la commande « ipconfig » dans cmd Windows pour chercher l'IP de notre machine Windows :



```
Windows [Version 10.0.19043.1052]
(c) Microsoft Corporation. All rights reserved.

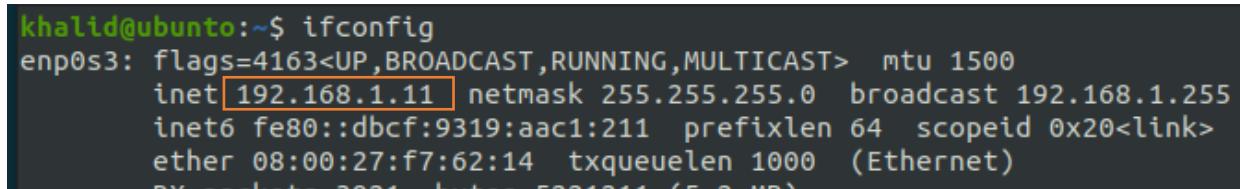
C:\Users\hp>ipconfig

Wireless LAN adapter Wi-Fi:

Connection-specific DNS Suffix . : Home
Link-local IPv6 Address . . . . . : fe80::1919:8ec3:1d7e:eb91%21
IPv4 Address. . . . . : 192.168.1.3
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : fe80::1%21
                           192.168.1.1
```

- Dans la machine Ubuntu :

On tape la commande « ifconfig » :



```
khalid@ubuntu:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.1.11 netmask 255.255.255.0 broadcast 192.168.1.255
            ether 08:00:27:f7:62:14 txqueuelen 1000 (Ethernet)
      inet6 fe80::dbcf:9319:aac1:211 prefixlen 64 scopeid 0x20<link>
```

Pour vérifier la connexion entre les deux machines :

On peut baser sur la commande « ping IP »

Dans windows on tape ping Linux IP:

```
C:\Users\hp>ping 192.168.1.11

Pinging 192.168.1.11 with 32 bytes of data:
Reply from 192.168.1.11: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.1.11:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Et pour linux on entre ping IP windows

```
khalid@ubunto:~$ ping 192.168.1.3
PING 192.168.1.3 (192.168.1.3) 56(84) bytes of data.
64 bytes from 192.168.1.3: icmp_seq=1 ttl=128 time=0.263 ms
64 bytes from 192.168.1.3: icmp_seq=2 ttl=128 time=0.300 ms
64 bytes from 192.168.1.3: icmp_seq=3 ttl=128 time=1.11 ms
64 bytes from 192.168.1.3: icmp_seq=4 ttl=128 time=0.301 ms
64 bytes from 192.168.1.3: icmp_seq=5 ttl=128 time=0.438 ms
64 bytes from 192.168.1.3: icmp_seq=6 ttl=128 time=0.324 ms
64 bytes from 192.168.1.3: icmp_seq=7 ttl=128 time=0.308 ms
```

⇒ On constate que les deux machines sont bien connectées.

Qu'allons-nous superviser ?

- Notre test va focaliser sur le contrôle du stockage de la machine windows de façon suivante : Lorsque l'espace plein d'un disque dur dépasse 90% il nous donne une notification que le service « Memory usage » est en mode critique (**critical**). Nagios va contrôler aussi le stockage de disque local tel que s'il dépasse 80% il nous détecter un « **warning** », et « **critical** » s'il dépasse 90% de stockage total.

Les plugins :

Dans ce test, on va dépendre majoritairement des plugins et de leur fonctionnement. Le caractère open source de nagios et sa large communauté ont donné la création de plusieurs plugins qui facilitent énormément la supervision.

Les plugins servent généralement à la récupération des données de nos hôtes (PC, routeurs, serveurs), et ils assurent la supervision à distance. Un certain

plugin a seulement l'accès à un certain type d'informations, puisqu'ils n'utilisent pas tous un même protocole pour communiquer, on ne peut pas donc s'appuyer totalement sur un seul plugin pendant tout notre processus de supervision

Par conséquent, pour superviser notre système d'informations tout entier avec toutes ses informations disponibles, on exploitera les plugins suivants :

check_disk : Vérifie l'espace occupé d'un disque dur

check_http : Vérifie le service "http" d'un hôte

check_ftp : Vérifie le service "ftp" d'un hôte

check_mysql : Vérifie l'état d'une base de données MYSQL

check_nt : Vérifie les différentes informations (disque dur, processeur ...) sur un système d'exploitation Windows.

check_nrpe: Permet de récupérer différentes informations sur les hôtes

check_ping: Vérifie la présence d'un équipement, ainsi que sa durée de réponse

check_pop: Vérifie l'état d'un service POP (serveur mail)

check_snmp : Récupère divers informations sur un équipement grâce au protocole SNMP (Simple Network Management Protocol)

Il est possible de créer son propre plugin. Dans ce cas, il faudra les créer de la sorte que celui renvoie à Nagios :

L'état du résultat (**OK, CRITICAL, DOWN, UP, ...**)

```

root@ubuntu:/usr/local/nagios/libexec# ls
check_apt           check_jabber          check_simap
check_asterisk.pl   check_load           check_sip
check_asterisk_sip_peers.sh check_log           check_smtp
check_breeze         check_mrtg            check_snmp
check_by_ssh         check_mrtgtraf        check_spop
check_clamd          check_nagios          check_ssh
check_cluster        check_netstat.pl      check_ssl_validity
check_cpu_stats.sh  check_nt              check_ssntp
check_dhcp           check_nntp            check_swap
check_dig             check_ntp             check_tcp
check_disk            check_nrpe            check_time
check_disk_smb        check_nt              check_udp
check_dns             check_ntp_peer        check_ups
check_dummy           check_ntp_time        check_uptime
check_file_age       check_ntp_overcr      check_users
check_flexlm          check_ntstat          check_wave
check_ftp             check_open_files.pl  check_yum
check_hpjd            check_oracle          custom_check_mem
check_http            check_overcr          custom_check_procs
check_icmp           check_ping            eventhandlers
check_ide_smart       check_pop             nagisk.pl
check_ifoperstatus    check_procs          negate
check_ifstatus        check_real            remove_perfdata
check_imap            check_rpc             urlize
check_init_service    check_sensors         utils.pm
check_ircd           check_services        utils.sh
root@ubuntu:/usr/local/nagios/libexec# 

```

Plugins retenus :

Après avoir consulté les différents plugins existants, nous avons choisi ceux qui correspondaient à notre cahier des charges.

Nous avons retenus les plugins suivants :

- **check_nt**
- **check_nrpe**
- **check_snmp**
- **check_ping**

⇒ [Check_nt](#)

Le plugin Check_nt est un plugin récent qui permet de superviser très facilement des PC dont le système d'exploitation est Windows.

Check_nt permet de récupérer sur un système Windows les informations suivantes : L'espace occupé sur le disque dur, le temps depuis le démarrage de l'ordinateur, la version du plugin NsClient ++, occupation du processeur, occupation de la mémoire, état d'un service.

Mise en place de check_nt :

1/On installe le plugin check_nt sur notre machine nagios. (ça se fait automatiquement avec l'installation des plugins)

2/ Sur notre hôte windows à superviser, on doit installer le logiciel NsClient++, téléchargeable sur le site <http://www.nsclient.org/download/>

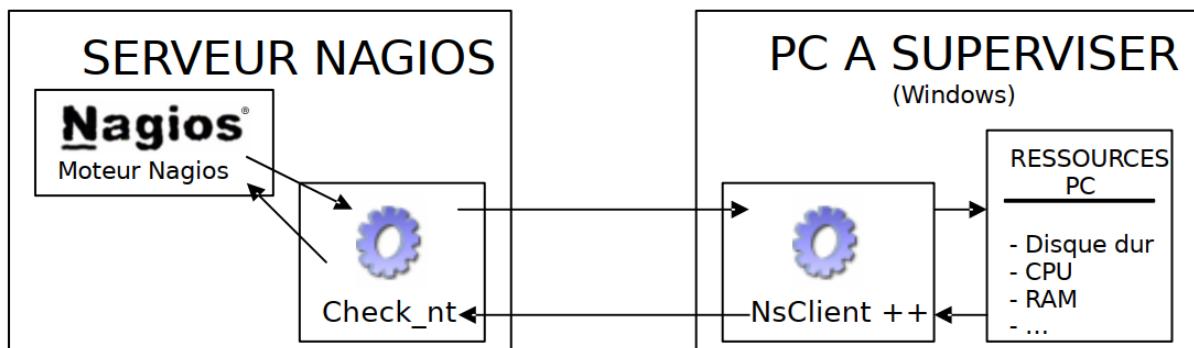
3/ Sur les machines à superviser, on doit configurer le fichier "NSC.ini".

C'est dans ce fichier que l'on doit définir :

- Le port sur lequel NsClient++ doit écouter les requêtes
- Les adresses des machines qui ont le droit de dialoguer avec NsClient++ (les machines qui ont le droit de récupérer les informations sur ce poste)
- Un mot de passe (les machines qui souhaiteront dialoguer avec celle-ci par NsClient++ devront fournir ce mot de passe)

=> Le fichier de configuration est fourni en annexe :

Fonctionnement de check_nt :



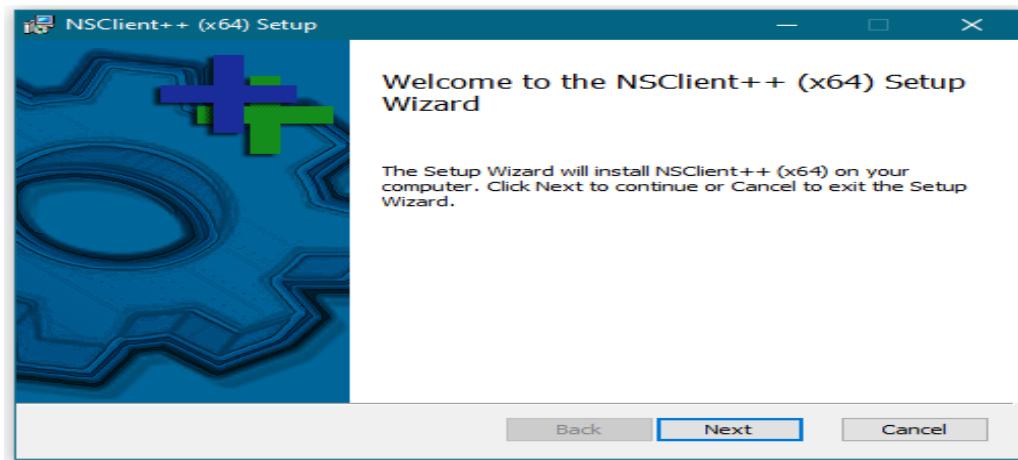
Lorsque NAGIOS veut récupérer les informations sur l'état d'un PC, il exécute le plugin **check_nt**. Celui-ci envoie une requête au PC supervisé. Sur le PC, le programme NsClient++ reçoit la requête, va chercher les informations dans les ressources du PC et renvoie le résultat au serveur Nagios.

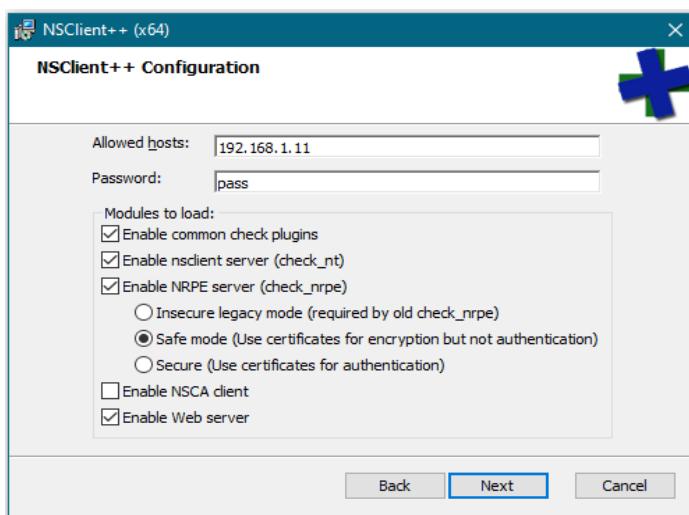
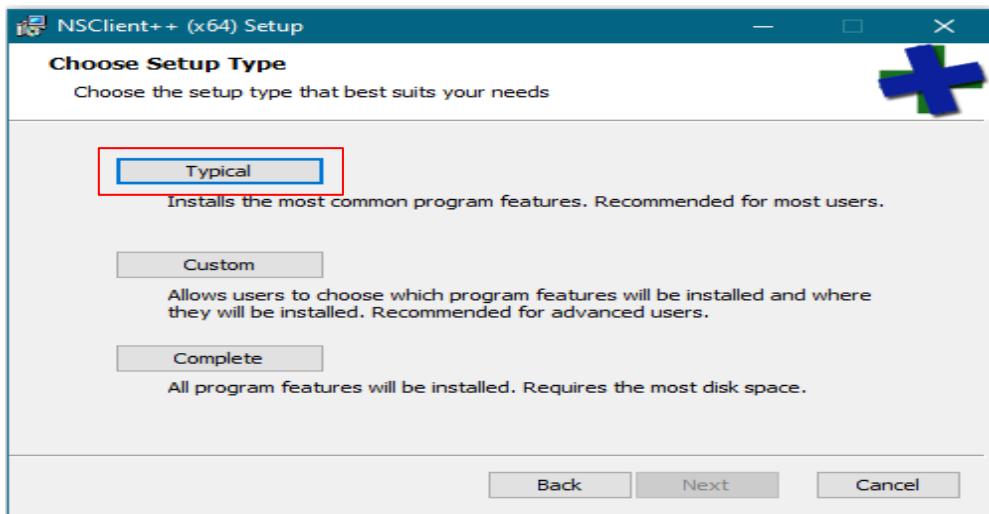
Installation du NsClient++ sur Windows :

0.5.2.35

See [github](#) for details

Feel free to [browse](#) other 0.5.2 releases

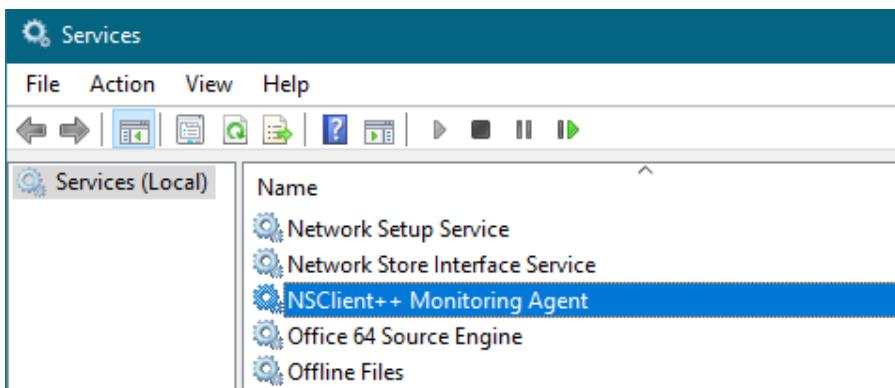


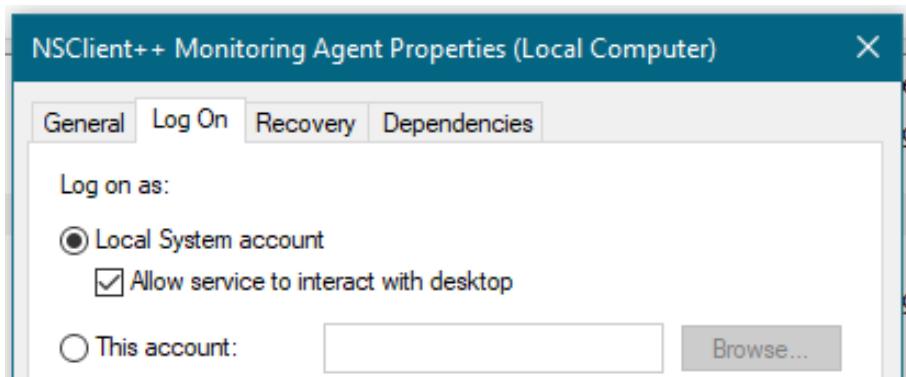


allowed hosts : l'adresse ip de la machine virtuel ubuntu.

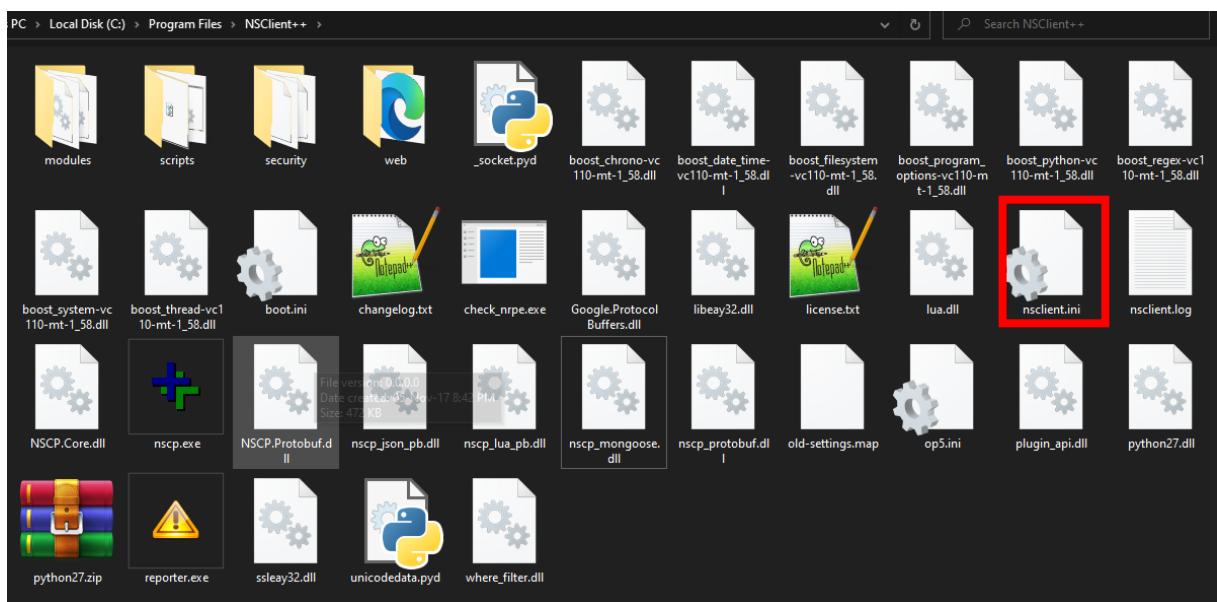
password: un mot de passe pour NSClient++

il faut modifier les paramètres du service Nsclient++ Monitoring agent.





On entre dans le fichier « C:\Program Files\NSClient++\nsclient.ini »



Pour rendre tous les modules enabled

```

8 ; in flight - TODO
9 [/settings/default]
10
11 ; Undocumented key
12 password = 2021
13
14 ; Undocumented key
15 allowed hosts = 192.168.1.30
16
17
18 ; in flight - TODO
19 [/modules]
20
21 ; Undocumented key
22 CheckExternalScripts = 1
23
24 ; Undocumented key
25 CheckHelpers = 1
26
27 ; Undocumented key
28 CheckEventLog = 1
29
30 ; Undocumented key
31 CheckNSCP = 1
32
33 ; Undocumented key
34 CheckDisk = 1
35
36 ; Undocumented key
37 CheckSystem = 1
38
39 ; Undocumented key
40 NSClientServer = enabled
41

```

Configuration du host et des services dans Nagios :

On doit se situer dans le répertoire **/usr/local/nagios/etc** pour créer les deux répertoires **hosts** et **services** qui va contenir tous les hôtes et leurs services qu'on va ajouter

```
root@ubuntu:~# cd /usr/local/nagios/etc  
root@ubuntu:/usr/local/nagios/etc# mkdir hosts services
```

```
root@ubuntu:/usr/local/nagios/etc# ls  
cgi.cfg      hosts          nagios.cfg  nrpe.cfg  printers    routers  services  
commands.cfg htpasswd.users nrpe        objects   resource.cfg servers  switches  
root@ubuntu:/usr/local/nagios/etc#
```

il faut les configurer dans le fichier de configuration nagios.cfg

```
root@ubuntu:/usr/local/nagios/etc# ls  
cgi.cfg      hosts          nagios.cfg  nrpe.cfg  printers    routers  services  
commands.cfg htpasswd.users nrpe        objects   resource.cfg servers  switches  
root@ubuntu:/usr/local/nagios/etc# vim nagios.cfg  
  
# directive as shown below:  
cfg_dir=/usr/local/nagios/etc/hosts  
cfg_dir=/usr/local/nagios/etc/services  
#cfg_dir=/usr/local/nagios/etc/servers  
#cfg_dir=/usr/local/nagios/etc/printers  
#cfg_dir=/usr/local/nagios/etc/switches  
#cfg_dir=/usr/local/nagios/etc/routers
```

51,1

La création de notre host Windows dans le rep hosts

```
root@ubuntu:/usr/local/nagios/etc# cd hosts  
root@ubuntu:/usr/local/nagios/etc/hosts# vim khalid.cfg
```

La modification du fichier Name_host.cfg par la commande « **sudo vim khalid.cfg** »

On change le nom de la machine Windows et son adresse IP dans la syntaxe **define host** :

```
define host{
    name linux-box ; Name of this template
    use generic-host ; Inherit default values
    check_period 24x7
    check_interval 5
    retry_interval 1
    max_check_attempts 10
    check_command check-host-alive
    notification_period 24x7
    notification_interval 30
    notification_options d,r
    contact_groups admins
    register 0 ; DONT REGISTER THIS - ITS A TEMPLATE
}

define host{
    use linux-box ; Inherit default values from a template
    host_name khalid ; The name we're giving to this server
    alias windows ; A longer name for the server
    address 192.168.1.3 ; IP address of Remote windows host
}
```

On fait un redémarrage de Nagios pour que ses informations soient enregistrées par la commande « **service nagios restart** » et « **service nagios status** » pour vérifier l'état de notre serveur nagios après les modifications. (On va faire ce test après chaque modification)

```
root@ubuntu:/usr/local/nagios/etc/hosts# service nagios restart
root@ubuntu:/usr/local/nagios/etc/hosts# service nagios status
● nagios.service - Nagios
   Loaded: loaded (/etc/systemd/system/nagios.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2021-06-29 13:11:01 +01; 4s ago
     Main PID: 4012 (nagios)
        Tasks: 6 (limit: 3490)
       Memory: 2.3M
      CGroup: /system.slice/nagios.service
              └─4012 /usr/local/nagios/bin/nagios /usr/local/nagios/etc/nagios.cfg
                  ├─4013 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios>
                  ├─4014 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios>
                  ├─4015 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios>
                  ├─4016 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios>
                  └─4017 /usr/local/nagios/bin/nagios /usr/local/nagios/etc/nagios.cfg
```

Current Network Status
Last Updated: Tue Jun 29 13:11:18 +01 2021
Updated every 90 seconds
Nagios® Core™ 4.4.5 - www.nagios.org
Logged in as nagiosadmin

Host Status Totals

Up	Down	Unreachable	Pending
2	0	0	0

All Problems All Types

0	2
---	---

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
9	0	7	0	0

All Problems All Types

7	16
---	----

Host Status Details For All Host Groups

Host	Status	Last Check	Duration	Status Info
khalid	UP	06-29-2021 13:09:44	0d 0h 51m 34s	PING OK - P
localhost	UP	06-29-2021 13:06:14	2d 0h 32m 52s	PING OK - P

Alors notre hôte (Khalid) est bien ajouté dans le serveur Nagios.

La création des services :

```
root@ubuntu:/usr/local/nagios/etc/hosts# cd ..
root@ubuntu:/usr/local/nagios/etc# cd services
root@ubuntu:/usr/local/nagios/etc/services# vim khalid.cfg
```

Service qu'on trouve par défaut :

```
define service{
use generic-service
host_name khalid
service_description check-host-alive
check_command check-host-alive
}
```

Création d'un service permettant de superviser la version de NSClient++

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

Création d'un service permettant de superviser le temps depuis la connexion de l'ordinateur

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

Création d'un service permettant de superviser la charge du processeur

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

Création d'un service permettant de superviser l'occupation de la mémoire

```
define service{
    use                  generic-service
    host_name           khalid
    service_description Memory Usage
    check_command       check_nt!MEMUSE!-w 80 -c 90
}
```

Création d'un service permettant de superviser l'espace occupé sur le disque dur

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

Création d'un service permettant de superviser les états de service

```
define service{
    use                  generic-service
    host_name           khalid
    service_description W3SVC
    check_command       check_nt!SERVICESTATE!-d SHOWALL -l W3SVC
}
```

Création d'un service permettant de superviser les processus en cours

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

Configuration de la commande check_nt :

De manière à vérifier la configuration de NSClient++ et que la communication entre Nagios et lui fonctionne, nous allons utiliser le plugin **check_nt** avec la variable **CLIENTVERSION**. Celle-ci permet de récupérer la version NSClient++ installée sur notre hôte Windows :

```
root@ubuntu:/usr/local/nagios/etc/services# /usr/local/nagios/libexec/check_nt -H 192.168.1.3 -p 12489 -v CLIENTVERSION -s pass
NSClient++ 0.5.2.35 2018-01-28
root@ubuntu:/usr/local/nagios/etc/services#
```

Puis on configure la commande check_nt dans le fichier commands.cfg

```
fatima@fatima-VirtualBox:/usr/local/nagios/etc/objects$ sudo vim commands.cfg
```

```
define command {
    command_name  check_nt
    command_line   $USER1$/check_nt -H $HOSTADDRESS$ -p 12489 -s nsclient -v $ARG1$ $ARG2$
}
```

L'ajout est le -s nsclient (mot de passe de notre NSClient++) et le - p le port sur lequel il faut envoyer la requête.

Connexion du Serveur Nagios depuis la machine Windows :

On se connecte par l'adresse :

<http://192.168.1.11/nagios/>

The screenshot shows the Nagios web interface with the following details:

- Current Network Status:** Last updated: Mon Jun 28 14:04:44 +01 2021. Updated every 30 seconds. Nagios® Core™ 4.4.5 - www.nagios.org. Logged in as nagiosadmin.
- Host Status Totals:** Up 2, Down 0, Unreachable 0, Pending 0. All Problems 0, All Types 2.
- Service Status Totals:** Ok 16, Warning 0, Unknown 0, Critical 0, Pending 0. All Problems 0, All Types 16.
- Service Status Details For All Hosts:** A table showing service status for hosts khalid and localhost. Services include C:\ Drive Space, CPU Load, Explorer, Memory Usage, NSClient++ Version, Uptime, W3SVC, check-host-alive, Current Load, Current Users, HTTP, PING, Root Partition, SSH, Swap Usage, and Total Processes. Most services are OK except for a few which are Pending.
- Limit Results:** Set to 100.
- Results:** 1 - 16 of 16 Matching Services.

Lors du test effectué par l'un des membres du groupe, toutes les vérifications effectuées pour son hôte supervisé S'avèrent être en état « OK » ce qui montre qu'aucune erreur ni anomalie n'apparaît sur sa machine.

Cependant, dans ce deuxième cas, la machine d'un autre membre du groupe relève des alertes retournés par les plugins selon les configurations effectuées :

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	06-29-2021 23:31:22	7d 3h 2m 32s	1/4	OK - load average: 0.47, 0.39, 0.39
	Current Users	OK	06-29-2021 23:32:25	7d 3h 1m 54s	1/4	USERS OK - 1 users currently logged in
	HTTP	OK	06-29-2021 23:28:27	7d 3h 1m 17s	1/4	HTTP OK: HTTP/1.1 200 OK - 11192 bytes in 0.002 second response time
	PING	OK	06-29-2021 23:29:30	7d 3h 0m 39s	1/4	PING OK - Packet loss = 0%, RTA = 0.12 ms
	Root Partition	OK	06-29-2021 23:30:20	7d 3h 0m 2s	1/4	DISK OK - free space: / 23576 MiB (62.84% inode=89%)
	SSH	OK	06-29-2021 23:30:20	3d 22h 56m 34s	1/4	SSH OK - OpenSSH_8.2p1 Ubuntu-4ubuntu1.2 (protocol 2.0)
	Swap Usage	OK	06-29-2021 23:30:20	7d 2h 58m 47s	1/4	SWAP OK - 100% free (1873 MB out of 1873 MB)
	Total Processes	OK	06-29-2021 23:30:20	7d 2h 58m 9s	1/4	PROCS OK: 45 processes with STATE = RSZDT
	C:\ Drive Space	CRITICAL	06-29-2021 23:31:43	1d 4h 40m 16s	3/3	c: - total: 115.85 Gb - used: 112.17 Gb (97%) - free 3.68 Gb (3%)
win-host	CPU Load	OK	06-29-2021 23:27:15	0d 1h 45m 58s	1/3	CPU Load 25% (5 min average)
	Explorer	CRITICAL	06-29-2021 23:23:48	1d 4h 41m 48s	3/3	Explorer.exe: not running
	Memory Usage	WARNING	06-29-2021 23:25:00	0d 1h 38m 13s	3/3	Memory usage: total:15006.53 MB - used: 12632.47 MB (84%) - free: 2374.06 MB (16%)
	NSClient++ Version	OK	06-29-2021 23:31:35	0d 1h 41m 38s	1/3	NSClient++ 0.5.2.35 2018-01-28
	PING	OK	06-29-2021 23:26:55	1d 5h 26m 8s	1/3	PING OK - Packet loss = 0%, RTA = 1.16 ms
	Uptime	OK	06-29-2021 23:32:28	0d 1h 40m 45s	1/3	System Uptime - 6 day(s) 4 hour(s) 36 minute(s)
	W3SVC	UNKNOWN	06-29-2021 23:29:10	0d 1h 44m 3s	3/3	Failed to open service W3SVC: 424: Le service spécifié n'existe pas en tant que service installé.
	check-host-alive	OK	06-29-2021 23:32:04	1d 21h 47m 18s	1/3	PING OK - Packet loss = 0%, RTA = 1.03 ms

Le service c:\ drive Space est en mode critical car le stockage dépasse 90%

Host	Service	Status	Last Check	Duration	Attempt	Status Information
win-host	Total Processes	OK	06-29-2021 23:40:20	7d 3h 8m 52s	1/4	PROCS OK: 45 processes with STATE = RSZDT
	C:\ Drive Space	WARNING	06-29-2021 23:41:43	0d 0h 2m 13s	3/3	c: - total: 115.85 Gb - used: 97.64 Gb (84%) - free 18.21 Gb (16%)
	CPU Load	OK	06-29-2021 23:37:15	0d 1h 56m 41s	1/3	CPU Load 30% (5 min average)
	Explorer	CRITICAL	06-29-2021 23:33:43	1d 4h 52m 31s	3/3	Explorer.exe: not running
	Memory Usage	WARNING	06-29-2021 23:35:00	0d 1h 48m 56s	3/3	Memory usage: total:15006.53 MB - used: 12730.12 MB (85%) - free: 2276.41 MB (15%)
	NSClient++ Version	OK	06-29-2021 23:41:35	0d 1h 52m 21s	1/3	NSClient++ 0.5.2.35 2018-01-28
	PING	OK	06-29-2021 23:36:55	1d 5h 36m 51s	1/3	PING OK - Packet loss = 0%, RTA = 0.85 ms
	Uptime	OK	06-29-2021 23:42:28	0d 1h 51m 28s	1/3	System Uptime - 6 day(s) 4 hour(s) 46 minute(s)
	W3SVC	UNKNOWN	06-29-2021 23:39:10	0d 1h 54m 46s	3/3	Failed to open service W3SVC: 424: Le service spécifié n'existe pas en tant que service installé.
	check-host-alive	OK	06-29-2021 23:42:04	1d 21h 58m 1s	1/3	PING OK - Packet loss = 0%, RTA = 0.64 ms

Mais lorsque on libère un peu d'espace (dépasse 80%) il passe le service passe en mode warning

Si on libère plus d'espace (moins de 80%), il va passer au mode ok :

Host	Service	Status	Last Check	Duration	Attempt	Status Information
win-host	Total Processes	OK	06-29-2021 23:30:20	7d 3h 1m 50s	1/4	PROCS OK: 45 processes with STATE = RSZDT
	C:\ Drive Space	OK	06-29-2021 23:51:43	0d 0h 1m 11s	1/3	c: - total: 115.85 Gb - used: 87.64 Gb (76%) - free 28.21 Gb (24%)
	CPU Load	OK	06-29-2021 23:47:15	0d 2h 5m 39s	1/3	CPU Load 31% (5 min average)
	Explorer	CRITICAL	06-29-2021 23:43:48	1d 5h 1m 29s	3/3	Explorer.exe: not running
	Memory Usage	WARNING	06-29-2021 23:45:00	0d 1h 57m 54s	3/3	Memory usage: total:15006.53 MB - used: 13027.20 MB (87%) - free: 1979.33 MB (13%)
	NSClient++ Version	OK	06-29-2021 23:51:35	0d 2h 1m 19s	1/3	NSClient++ 0.5.2.35 2018-01-28
	PING	OK	06-29-2021 23:46:55	1d 5h 45m 49s	1/3	PING OK - Packet loss = 0%, RTA = 1.03 ms
	Uptime	OK	06-29-2021 23:52:28	0d 2h 0m 26s	1/3	System Uptime - 6 day(s) 4 hour(s) 56 minute(s)
	W3SVC	UNKNOWN	06-29-2021 23:49:10	0d 2h 3m 44s	3/3	Failed to open service W3SVC: 424: Le service spécifié n'existe pas en tant que service installé.
	check-host-alive	OK	06-29-2021 23:52:04	1d 22h 6m 59s	1/3	PING OK - Packet loss = 0%, RTA = 0.99 ms

Mais le service memory usage ne change pas car il compte sur le stockage total

Host	Service	Status	Last Check	Duration	Attempt	Status Information
win-host	C:\ Drive Space	OK	06-30-2021 05:11:43	0d 5h 24m 49s	1/3	c: - total: 115.85 Gb - used: 88.27 Gb (76%) - free 27.57 Gb (24%)
	CPU Load	OK	06-30-2021 05:10:17	0d 7h 29m 17s	1/3	CPU Load 32% (5 min average)
	Explorer	CRITICAL	06-30-2021 05:13:48	1d 10h 25m 7s	3/3	Explorer.exe: not running
	Memory Usage	OK	06-30-2021 05:15:00	0d 4h 21m 32s	1/3	Memory usage: total:16038.68 MB - used: 13737.47 MB (86%) - free: 2301.41 MB (14%)
	NSClient++ Version	OK	06-30-2021 05:11:35	0d 7h 24m 57s	1/3	NSClient++ 0.5.2.35 2018-01-28
	PING	OK	06-30-2021 05:11:19	1d 11h 9m 27s	1/3	PING OK - Packet loss = 0%, RTA = 1.14 ms
	Uptime	OK	06-30-2021 05:12:28	0d 7h 24m 4s	1/3	System Uptime - 6 day(s) 10 hour(s) 16 minute(s)
	W3SVC	UNKNOWN	06-30-2021 05:09:10	0d 7h 27m 22s	3/3	Failed to open service W3SVC: 424: Le service spécifié n'existe pas en tant que service installé.
	check-host-alive	OK	06-30-2021 05:12:04	2d 3h 30m 37s	1/3	PING OK - Packet loss = 0%, RTA = 0.97 ms

Conclusion :

Ce travail effectué dans le cadre d'un projet académique nous a permis généralement de s'avancer au niveau de l'administration d'un système sous Linux (UBUNTU), et particulièrement de Comprendre les concepts de base de la supervision dans un système d'information et de mettre en évidence les différentes architectures possibles dans le cadre de la supervision à l'aide du logiciel Nagios.

Après le processus de l'installation, la configuration et le test de cet outil de configuration. On constate déjà jusqu'à quel point la supervision continue facilite la tâche d'un administrateur. En effet, on arrive à trouver et à détecter des erreurs et des anomalies aisément. Ce qui aurait pu prendre beaucoup du temps et d'effort si on devait le faire à la main. Et d'ailleurs, on devait anticiper que ce genre d'erreurs vont apparaître.

Il ne reste maintenant que la correction de ces problèmes qui est laissé à l'expertise de l'administrateur.

Dictionnaire de commande :

- **Apt** : Advanced Packaging Tool, elle est utilisée pour l'installation des paquets pour un nouveau logiciel, la mise à jour et l'actualisation.
- **Autoconf** : c'est un paquet qui traverse le système puis crée un script de configuration pour un certain projet
- **Bc** : la commande bc pour basic calculator est faite pour faire les opérations arithmétiques basiques
- **Gawk** : pour GNU awk, c'est donc l'outil de ubuntu qui sert à implémenter le langage awk. Ce dernier est le langage qui a été fait pour UNIX dans ses débuts et qui sert à traiter les données qui sont surtout organisés sous forme de colonnes.
- **Dc** : un outil pour soutenir la précision arithmétique illimitée.
- **build-essentials** : c'est un meta-package (il n'installe rien mais il fait appel à d'autres paquets), il fait installer des outils de compilations et de débogage, ainsi que des bibliothèques gcc et g++.
- **Gcc** : compilateur c++.
- **libc6** : contient des fichier pour exécuter des softwares faits avec C.
- **wget** : une commande qui nous permet de faire un téléchargement d'un lien externe web à partir de notre système ububntu.
- **Unzip** : est un outil d'extraction
- **apache2** : est un serveur web utilisé couramment sur linux pour fournir les pages demandées aux clients web (navigateurs).
- **Libapache2-mod-php** : un paquet de dépendance, il fournit le module PHP pour le serveur web apacha2.
- **Libgd-dev** : un paquet de dépendance, il contient une bibliothèque graphique dans sa version de développement.

- **Libmcrypt-dev** : c'est une bibliothèque de décryptage qui contient des algorithmes pour décrypter du contenu crypté par mcrypt (outil de cryptage de fichiers).
- **Libssl-dev** : (SSL : Secure Socket Layer), c'est un paquet qui contient des protocoles pour sécuriser la communication via Internet en s'appuyant sur le cryptage.
- **Snmp** : (Simple Network Management Protocole) est un protocole de communication qui permet aux administrateurs réseau de gérer les équipements du réseau, de superviser et de diagnostiquer des problèmes réseaux et matériels à distance.
- **Libnet-snmp-perl** : le libnet est un module qui sert à implémenter une interface orientée objet pour le protocole snmp. Perl est une famille de langages de programmation de haut niveau qui sont utilisés pour administrer. Des applications perl utilisent ce module pour traiter des informations à distance.
- **gettext**: est une bibliothèque dont le but est d'internationaliser, i.e d'écrire des programmes multilingues.
- **tar** : Cette commande est utilisée pour compresser ou extraire des fichiers, elle peut également être utilisée pour la modification de documents tar.
- **Tar –xvf** : x(extract)□ extraire , v (verbose)□ créer explicitement les fichiers extraits, f(file)□ le fichier à suivre est celui à extraire
- **./configue** : on vérifie si on a tout dont on a besoin pour installer l'application.
- **-- (double dash)** : est utilisée pour montrer que la commande et ses spécifications sont terminées. Et que ce qui va suivre représente des arguments.
- **Make** : elle cherche les fichiers qui doivent être compilés et elle émet les commandes qui sont nécessaires pour le faire.

- **Usermod** : modify user, c'est une commande qui sert à changer les attributs d'un certain utilisateur.
- **Option-a** : Ajouter l'utilisateur aux groupes supplémentaires. Cette option ne peut être utilisé qu'avec l'option –G
- **Option -G** : Liste de groupes supplémentaires auxquels appartient également l'utilisateur.
- **Apache2** : l'appellation de debian et Ubuntu pour le serveur apache
- **A2enmod rewrite** : cette commande permet d'activer le mod_rewrite sur Apache2 qui permet de gérer la ré-écriture d'URL.
- **htpasswd** : cette commande est utilisée pour la mise à jour des fichiers contenant les données d'authentification d'un site web en ajoutant des données ou en modifiant d'autres existantes. Ne fonctionne qu'avec les flat-files, des fichiers en deux dimensions.
- **Systemctl** : cette commande est l'outil de gestion essentiel pour contrôler le système d'initialisation. Elle permet de gérer les services, vérifier les états, modifier les états du système et travailler avec les fichiers de configuration.
- **Systemctl restart** : cette commande permet de redémarrer le service Nagios et Apache2 qui sont en cours d'exécution.
- **Systemctl enable** : permet d'activer le service de démarrage Nagios et apache2.
- **Systemctl status** : permet de vérifier l'état du service Nagios et Apache2 sur notre système.
- **Ifconfig** : cette commande affiche les informations des interfaces réseau IP à partir de l'interpréteur de commandes.
- **Macros** : une seule instruction qui se développe automatiquement en un ensemble d'instructions pour effectuer une tâche particulière.
- **User Macros** : sont utiles si vous souhaitez créer vos propres macros personnalisées. Il peut s'agir d'effectuer des actions spécifiques, d'appliquer un formatage personnalisé et bien plus encore.
- **ll** : lister le contenu d'un répertoire

- **Sudo vim contacts.cfg:** cette commande permet de modifier le fichier de configuration contacts.cfg, pour définir l'email de l'administrateur nagios.
Pour cela on remplace l'email existant (nagios@localhost) par notre email et on l'enregistre.
- **Sudo vim commands.cfg:** cette commande permet de modifier le fichier de configuration commands.cfg, pour définir le **nrpe**
- **Nrpe:** Nagios Remote Plugin Executor: est un agent Nagios qui permet la surveillance du système à distance à l'aide de scripts hébergés sur les systèmes distants. Il permet de surveiller les ressources telles que l'utilisation du disque, la charge du système ou le nombre d'utilisateurs actuellement connectés.
- **UFW:** Son objectif principal est de simplifier la gestion de votre **firewall** et de fournir une interface facile à utiliser. Il est bien pris en charge et populaire dans la communauté Linux, même installé par défaut dans de nombreuses distributions.
- **Sudo ufw status:** cette commande permet de vérifier si ufw fonctionne.
- **Sudo ufw allow apache:** cette commande permet d'autoriser le serveur apache au UFW firewall.
- **Sudo ufw status:** on a utilisé cette commande une autre fois après la commande (sudo ufw allow apache) pour vérifier si la permission **apache** a été ajoutée.
- **Sudo ufw reload:** permet de relancer le UFW qui devrait à terme permettre une configuration automatique du firewall lors de l'installation de programmes en ayant besoin.

Bibliographie :

- 1.<https://linux.101hacks.com/unix/check-ping/>
2. <https://support.nagios.com/>

3. <https://nsclient.org/download/>
4. www.monitoring-plugins.org
5. <https://wiki.monitoring-fr.org/nagios/installation-layout>
6. <https://askubuntu.com/>
7. <https://www.edureka.co/blog/nagios-tutorial/>
8. <https://www.ionos.fr/digitalguide/serveur/outils/nagios-une-vue-densemble-des-systemes-et-reseaux/>
9. www.howtoforge.com
10. https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3/html/console_administration_guide/configuring_nagios_to_send_mail_notification
11. <https://openclassrooms.com/fr/courses/2035786-mettez-en-place-un-outil-de-supervision-de-production-avec-nagios/6049763-quest-ce-que-la-supervision>
12. <https://www.youtube.com/watch?v=lsL1nA8BJwA>
13. <https://www.youtube.com/watch?v=EIWxbO3ogVs>
14. <https://www.youtube.com/watch?v=EIWxbO3ogVs>
15. <https://www.youtube.com/watch?v=EIWxbO3ogVs&>