

Procédure d'installation NAGIOS

L'objectif de cette procédure est de réaliser l'installation de NAGIOS sur Ubuntu Live server version 20.04.

Sommaire :

Configuration de départ :.....	1
Installation des paquets nécessaires au fonctionnement de NAGIOS :.....	1
Création de l'environnement de NAGIOS :.....	2
Installation de l'arborescence de NAGIOS :.....	2
Installation des plugins NAGIOS :.....	3
Création d'alias pour les commandes récurrentes :.....	3
Création d'un répertoire pour stocker les fichiers de configuration :.....	4
Création d'une commande/host/service :.....	4
Vérification de l'accès à l'utilisation de la commande « check_http » par l'utilisateur nagios :.....	5
Création d'une commande « check_http » spécifique :.....	6
Mise en place d'un service supervisé par Nagios :.....	6
Sur la machine du serveur Teamspeak :.....	6
Création des checks sur le serveur NAGIOS :.....	7

Configuration de départ :

- Installer une configuration minimale d'Ubuntu Live Server 20.04 en activant le serveur SSH lors de la configuration.
- Passer en mode root avec la commande « sudo -i »
- Effectuer les mises à jour des paquets avec les commandes « apt update » et « apt upgrade »

Installation des paquets nécessaires au fonctionnement de NAGIOS :

- Installer le serveur Apache, PHP et les autres paquets nécessaires avec la commande « apt-get install apache2 php php-gd php-imap php-curl »
- Installation des bibliothèques Perl avec la commande « apt-get install libxml-libxml-perl libnet-snmp-perl libperl-dev libnumber-format-perl libconfig-inifiles-perl libdatetime-perl libnet-dns-perl »
- Installation des bibliothèques graphiques avec la commande « apt-get install libpng-dev libjpeg-dev libgd-dev »

- Installation des outils de compilations standards avec la commande « apt-get install gcc make autoconf libc6 unzip »

Création de l'environnement de NAGIOS :

- Création d'un nouvel utilisateur pour l'utilisation de NAGIOS avec la commande « useradd -m nagios » puis la commande « passwd nagios » pour configurer un mot de passe pour cet utilisateur
- Création d'un groupe spécifique pour l'utilisation de commandes extérieures avec la commande « groupeadd nagcmd »
- Ajouter les utilisateurs nagios et www-data au groupe nagcmd avec les commandes « usermod -a -G nagcmd nagios » et « usermod -a -G nagcmd www-data »
- Création d'un répertoire pour regrouper les sources qui vont être utilisées dans la suite de cette procédure avec la commande « mkdir /home/nagios/downloads »
- Se rendre dans le répertoire downloads créé précédemment avec la commande « cd /home/nagios/downloads/ »
- Se rendre sur le site de NAGIOS puis trouver le téléchargement de NAGIOS et copier le lien afin de l'utiliser dans la commande suivante « wget https://go.nagios.org/get-core/4-4-13 »
- Possibilité de renommer le fichier en nagios-[numéro de version].tar.gz avec la commande « mv 4-4-13 nagios-4.4.13.tar.gz »
- Décompresser l'archive avec la commande « tar -zxvf nagios-4.4.13.tar.gz »
- Se rendre dans le dossier qui viens d'être décompressé avec la commande « cd nagios-4.4.13 »
- Installer mcrypt avec les commandes suivantes : « apt install -y build-essential », « apt install php php-pear php-dev libmcrypt-dev », « pecl search mcrypt » et « pecl install mcrypt »
- Utiliser la commande de configuration suivante « ./configure --with-httpd-conf=/etc/apache2/sites-enabled --with-command-group=nagcmd »
- Lancer la compilation avec la commande « make all »

Installation de l'arborescence de NAGIOS :

- Lancer l'installation avec la commande « make install »
- Installer le service NAGIOS avec la commande « make install-daemoninit »
- Installer le pipe de NAGIOS avec la commande « make install-commandmode »

- Installer les fichiers de configuration de NAGIOS avec la commande « make install-config »
- Installer l'interface d'administration web de NAGIOS avec la commande « make install-webconf »
- Activer les modules d'Apache nécessaires avec les commandes : « a2enmod rewrite », « a2enmod cgi »
- Configurer l'accès à Apache en changeant le couple identifiant/mot de passe pour s'y connecter avec la commande « httpasswd -cb /usr/local/nagios/etc/httpasswd.users nagiosadmin nagiosadmin »
- Configurer les droits d'accès sur le dossier nagios avec la commande « chown -R nagios:nagcmd /usr/local/nagios »
- Redémarrer les services apache et nagios avec les commandes « systemctl restart apache2 » et « systemctl restart nagios »
- Vérification de l'activation des services NAGIOS avec les commandes « ps -edf | grep nagios » et « systemctl status nagios »

Installation des plugins NAGIOS :

- Se rendre sur le site de NAGIOS (noagios.org) et aller dans l'onglet plugins puis copier le lien du fichier plugin pour l'utiliser dans la commande « wget https://nagios-plugins.org/download/nagios-plugins-2.4.6.tar.gz »
- Décompresser le fichier téléchargé avec la commande « tar zxvf nagios-plugins-2.4.6.tar.gz »
- Configurer les plugins avec la commande « ./configure --with-nagios-user=nagios --with-nagios-group=nagcmd » puis utiliser la commande « make »
- Installation des plugins avec la commande « make install »
- Vérifier l'installation des plugin avec la commande « ls -lrth /usr/local/nagios/libexec/ »

Création d'alias pour les commandes récurrentes :

- Éditer le fichier .bashrc de NAGIOS avec la commande « nano /home/nagios/.bashrc »
- Dans ce fichier ajouter la ligne suivante à la fin de celui-ci : « alias testnagios="/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg" »
- Changer d'utilisateur avec la commande « su - nagios » afin de tester l'alias et passer en shell bash en utilisant la commande « bash » une fois le changement d'utilisateur effectué (inutile si l'on utilise la commande ci-après pour changer définitivement le shell associé à l'utilisateur nagios)

- Utiliser la commande « `usermod --shell /bin/bash nagios` » afin d'être directement en shell bash à l'avenir
- Utiliser la commande « `testnagios` » afin de vérifier son bon fonctionnement
- Ajouter la commande suivante dans le fichier `sudoers` : « `echo "nagios ALL=NOPASSWD:/bin/systemctl restart nagios" >> /etc/sudoers` », cela permet à l'utilisateur nagios de relancer le service nagios avec la commande suivante : « `sudo systemctl restart nagios` »
- Créer un alias pour la commande précédente avec les commandes suivante : « `nano /home/nagios/.bashrc` » puis à la fin du fichier, ajouter la ligne « `alias restartnagios="sudo systemctl restart nagios"` »

Création d'un répertoire pour stocker les fichiers de configuration :

- Se rendre dans le répertoire de NAGIOS avec la commande « `cd /usr/local/nagios` »
- Créer un dossier dans lequel seront stockés les fichiers de configuration avec la commande « `mkdir configfolder` »
- Éditer le fichier de configuration de NAGIOS afin que celui-ci prenne en compte le dossier nouvellement créé avec la commande « `nano /usr/local/nagios/etc/nagios.cfg` »
- Se rendre tout en bas de la page et ajouter les lignes suivantes : « `#CONFIG FOLDER` » et « `cfg_dir=/usr/local/nagios/configfolder` »
- Tester le bon fonctionnement de NAGIOS avec la commande « `testnagios` » puis redémarrer le service NAGIOS avec la commande « `restartnagios` »

Création d'une commande/host/service :

- Se rendre dans le répertoire « `configfolder` » avec la commande « `cd /usr/local/nagios/configfolder` »
- Créer un fichier de configuration avec la commande « `touch nagios-server.cfg` »
- Modifier le contenu du fichier nouvellement créé avec la commande « `nano nagios-server.cfg` »
- Écrire les lignes suivantes dans le fichier de configuration :

```
define command {  
    command_name    check-ping-localhost  
    command_line    $USER1$/usr/local/nagios/libexec/check_ping -H 127.0.0.1 -w  
40,40% -c 60,60%
```

```
}
define command {
    command_name    check-ssh-localhost
    command_line    $USER1$/check_ssh 127.0.0.1
}
define host {
    host_name        Nagios Server
    address          localhost
    check_command    check-ping-localhost
    max_check_attempts 3
    contacts         nagiosadmin
}
define service {
    service_description SSH sur Nagios Server
    host_name        Nagios Server
    check_command    check-ssh-localhost
    max_check_attempts 3
    contacts         nagiosadmin
}
```

- Pour fixer l'accès aux commandes si Nagios dit qu'il n'a pas les permissions (dans l'interface web :

```
« chown nagios.nagcmd /usr/local/nagios/var/rw »
« chmod g+rwX /usr/local/nagios/var/rw »
« chmod g+s /usr/local/nagios/var/rw »
```

- Dans l'interface web les dates sont notées au format US (mois/jour/année) et à la timezone UTC0 (timezone de Londres)

- Se rendre dans le répertoire « configfolder » avec la commande « cd /usr/local/nagios/configfolder »

- Editer le fichier « nagios-server.cfg » avec la commande « nano nagios-server.cfg »

- Dans la section « define host » ajouter la ligne « parents localhost »

- Sauvegarder les modifications et effectuer les commandes « testnagios » et « restartnagios » puis se rendre sur l'interface web puis dans « map » pour afficher la carte du réseau supervisé par NAGIOS avec les changements effectués.

Vérification de l'accès à l'utilisation de la commande « check_http » par l'utilisateur nagios :

- Se rendre dans le répertoire « libexec » avec la commande « cd /usr/local/nagios/libexec »

- Exécuter la commande « ./check_http -H localhost » afin de vérifier le statut du localhost

- Vérifier le bon fonctionnement de l'interface de nagios avec la commande « `./check_http -H localhost -u /nagios/main.php -a nagiosadmin:nagiosadmin` » (possibilité de rajouter `-E` à la fin de la commande pour avoir des résultats plus détaillés)

Création d'une commande « `check_http` » spécifique :

- Se rendre dans le répertoire des fichiers de configuration avec la commande « `cd /usr/local/nagios/configfolder` »
- Éditer le fichier « `nagios-server.cfg` » avec la commande « `nano nagios-server.cfg` » et ajouter la commande suivante :

```
define command {
    command_name    check-http-spec
    command_line    $USER1$/check_http -H $HOSTADDRESS$ -w 0.1 -c 0.2 -a
$ARG1$ -u $ARG2$ -s $ARG3$ -m $ARG4$
}
define service {
    host_name        Nagios Server
    service_description HTTP sur Nagios Server
    check_command    check-http-spec!nagiosadmin:nagiosadmin!/nagios/main.php!'4.4.13!'
5000:10000
    max_check_attempts 3
    contacts         nagiosadmin
}
```

Mise en place d'un service supervisé par Nagios :

- Disposer d'un serveur NAGIOS paramétré avec la procédure ci-dessus
- Sélectionner un service à installer sur une VM (ici un serveur teamspeak) afin que celui-ci soit géré par NAGIOS (Le service sera installée sur une machine « vierge », il n'y aura rien d'autre dessus que le service que l'on souhaite administrer)
- Faire en sorte que les 2 machines communiquent entre elles (réseau privé hôte)

Sur la machine du serveur Teamspeak :

(- Suivre le tutoriel suivant pour installer un serveur teamspeak :
<https://skoali.com/help/vps/installer-un-serveur-teamspeak>)

- Installer le paquet w3m avec la commande « `apt install w3m` » afin de pouvoir ouvrir le portail de connexion du lycée avec la commande suivante « `w3m https://dne.lycees.bourgognefranchecomte.fr/auth/` »

- Utiliser la commande wget suivante afin de télécharger teamspeak server « wget https://files.teamspeak-services.com/releases/server/3.13.7/teamspeak3-server_linux_amd64-3.13.7.tar.bz2 »

```

-----
I M P O R T A N T
-----
Server Query Admin Account created
loginname= "serveradmin", password= "RC2GOP7v"
apikey= "BABej1TrZHg30P2xcZao30qKAJJhvI5BLYx7e8u"
-----
^C
root@tsserver:/home/ts/teamspeak3-server_linux_amd64#
-----
I M P O R T A N T
-----
ServerAdmin privilege key created, please use it to gain
serveradmin rights for your virtualserver. please
also check the doc/privilegekey_guide.txt for details.

token=g4KRCoo8R95PzZ4a+3frYc7Bh8k8QqPPX7oH4AlQ
-----

```

- Changer le propriétaire du dossier « ts » avec la commande « chown -R user:user teamspeak3-server_linux_amd64-3.13.7 » afin de pouvoir lancer le service à partir de l'utilisateur par défaut.
- Pour démarrer le serveur Teamspeak, utiliser la commande « /home/ts/teamspeak3-server_linux_amd64/ts3server_startscript.sh start » et pour le stopper utiliser la commande : « /home/ts/teamspeak3-server_linux_amd64/ts3server_startscript.sh stop »

Création des checks sur le serveur NAGIOS :

- Sur la machine NAGIOS, se rendre dans le répertoire « configfolder » avec la commande « cd /usr/local/nagios/configfolder »
- Modifier le contenu du fichier nouvellement créé avec la commande « nano nagios-server.cfg »
- Écrire les lignes suivantes dans le fichier de configuration :

```

define host {
    host_name          Serveur Teamspeak
    address             192.168.56.106
    check_command       check_ping!40,40%!60,60%
    max_check_attempts  3
    contacts            nagiosadmin
}
define service {
    host_name          Serveur Teamspeak

```

```

    service_description    TCP Serveur Teamspeak
    check_command          check_tcp!10011
    max_check_attempts     3
    contacts               nagiosadmin
}

```

- Tester le bon fonctionnement de NAGIOS avec la commande « testnagios » puis redémarrer le service NAGIOS avec la commande « restartnagios »

- Si tout fonctionne correctement, on obtient les résultats suivants :

Host	Status	Last Check	Duration	Status Information
Nagios Server	UP	01-22-2024 14:10:19	39d 6h 58m 51s	PING OK - Packet loss = 0%, RTA = 0.06 ms
Serveur Teamspeak	UP	01-22-2024 14:09:04	0d 0h 3m 12s	PING OK - Packet loss = 0%, RTA = 1.13 ms
localhost	UP	01-22-2024 14:07:49	59d 23h 21m 10s	PING OK - Packet loss = 0%, RTA = 0.05 ms

Host	Service	Status	Last Check	Duration	Attempt	Status Information
Nagios Server	HTTP sur Nagios Server	OK	01-22-2024 14:11:19	0d 1h 16m 56s	1/3	HTTP OK: HTTP/1.1 200 OK - 8486 bytes in 0.001 second response time
	SSH sur Nagios Serveur	OK	01-22-2024 14:12:19	42d 0h 4m 54s	1/3	SSH OK - OpenSSH_8.2p1 Ubuntu-4ubuntu0.9 (protocol 2.0)
Serveur Teamspeak	TCP Serveur Teamspeak	OK	01-22-2024 14:11:37	0d 0h 11m 38s	1/3	TCP OK - 0.001 second response time on 192.168.56.106 port 10011
localhost	Current Load	OK	01-22-2024 14:08:19	59d 23h 18m 24s	1/4	OK - load average: 0.00, 0.00, 0.00
	Current Users	OK	01-22-2024 14:09:19	53d 5h 50m 53s	1/4	USERS OK - 2 users currently logged in
	HTTP	OK	01-22-2024 14:10:19	42d 0h 46m 10s	1/4	HTTP OK: HTTP/1.1 200 OK - 11192 bytes in 0.000 second response time
	PING	OK	01-22-2024 14:11:49	59d 23h 21m 25s	1/4	PING OK - Packet loss = 0%, RTA = 0.04 ms
	Root Partition	OK	01-22-2024 14:12:49	59d 23h 20m 54s	1/4	DISK OK - free space: / 5098 MiB (46.87% inode=81%):
	SSH	OK	01-22-2024 14:08:49	59d 23h 20m 16s	1/4	SSH OK - OpenSSH_8.2p1 Ubuntu-4ubuntu0.9 (protocol 2.0)
	Swap Usage	OK	01-22-2024 14:09:49	59d 23h 19m 39s	1/4	SWAP OK - 100% free (2047 MB out of 2047 MB)
	Total Processes	OK	01-22-2024 14:12:00	59d 23h 19m 1s	1/4	PROCS OK: 42 processes with STATE = RSZDT