Annexe 2
Superviser un réseau avec Nagios

Sommaire :

I - Introduction

Nagios est une application open source permettant la surveillance système et réseau.

Nagios dispose d'une interface web intégrée dans laquelle nous pouvons accéder et surveiller l'ensemble de l'infrastructure en un seul endroit. Vous devez travailler au niveau du fichier (fichier de configuration) car vous ne pouvez pas personnaliser ou ajuster les paramètres de surveillance via l'interface Web.

Tout ce qui suit a été validé sur une machine virtuelle **Debian 10 Buster** 64 bits avec **nagios core 4.4.7**.

II - Installation de Nagios Core

II.1. Avant d'installer Nagios Core

Avant d'installer **Nagios Core**, il faut d'abord installer les paquets suivants :

```
apt-get update
apt-get -y install curl build-essential apache2 php openssl perl make php-gd
libgd-dev libapache2-mod-php libperl-dev libssl-dev daemon wget apache2-
utils unzip snmp snmpd libsnmp-dev
```

Remarque : Le paquet **libsnmp-dev** permettra de créer le plugin **check_snmp** lors de l'installation des **plugins** de **Nagios**.

Il faut créer les différents **utilisateurs** et **groupes** :

```
useradd nagios
groupadd nagcmd
usermod -a -G nagcmd nagios
usermod -a -G nagcmd www-data
```

II.2. Installation de Nagios Core

Puis on récupère **Nagios Core** (https://github.com/NagiosEnterprises/nagioscore/releases) :

```
cd /tmp
wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-
4.4.7.tar.gz
tar -zxvf nagios-4.4.7.tar.gz
cd /tmp/nagios-4.4.7/
```

On compile et on installe maintenant **Nagios Core** (L'installation se fait dans le répertoire **/usr/local/nagios**) :

```
./configure --disable-ssl --with-nagios-group=nagios --with-command-
group=nagcmd --with-httpd_conf=/etc/apache2/sites-enabled/
make all
make install
make install-init
make install-config
make install-commandmode
make install-webconf
```

Il faut créer un compte pour **Apache** pour se connecter à l'interface Web de **Nagios**. On créé le compte utilisateur " **nagiosadmin** " avec le mot de passe " **nagiosadmin** " à l'aide de la commande :

```
htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

Puis on exécute les commandes :

```
a2enmod cgi
service apache2 restart
```

II.3. Installation des plugins de Nagios

Il faut maintenant récupérer et installer les **plugins** de **Nagios** :

```
cd /tmp
wget https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz
tar -zxvf /tmp/nagios-plugins-2.3.3.tar.gz
cd /tmp/nagios-plugins-2.3.3/
./configure --with-nagios-user=nagios --with-nagios-group=nagios
```

make
make install

II.4. Configuration initiale de Nagios

Les fichiers de configuration sont dans le **répertoire** "/**usr/local/nagios/etc**". Vous n'avez pas besoin de modifier les fichiers de configuration de Nagios pour démarrer l'outil de surveillance Nagios. Tout ce dont vous avez besoin est de mettre à jour l'adresse e-mail dans le fichier " **/usr/local/nagios/etc/objects/contacts.cfg** " pour " **nagiosadmin** " avant de démarrer Nagios .

On peut vérifier la configuration de **Nagios** :
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

On peut maintenant lancer **Nagios** :
systemctl start nagios.service

Maintenant, on peut accéder à l'interface Web de **Nagios** en utilisant l'URL **http://ip-address/nagios/**. Le navigateur nous demande d'entrer le nom d'utilisateur " **nagiosadmin** " et le mot de passe que vous avez spécifié précédemment.

III - La supervision avec Nagios

III.1. Introduction

Pour configurer finement ce que l'on veut superviser, il faudra obligatoirement reprendre la configuration à la main en éditant les fichiers se trouvant dans le répertoire **/usr/local/nagios/etc/objects/**.

Si on rajoute des fichiers de configuration (par exemple **monpc.cfg**), il suffit de les mettre dans le dossier **/usr/local/nagios/etc/objects/monreso** et de le signaler à **Nagios** en modifiant le fichier **/usr/local/nagios/etc/nagios.cfg** de la façon suivante :
cfg_dir=/usr/local/nagios/etc/objects/monreso

Toute modification pourra être testée grâce à la commande :
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

III.2. Comment superviser un élément actif ?

Pour superviser un élément actif, par exemple le serveur «**serveur-linux**», il est nécessaire de disposer d'un fichier de configuration (par exemple *serveur-linux.cfg*) dans le répertoire ***/usr/local/nagios/etc/objects/monreso***.

La structure type d'un tel fichier commence par le mot clé "**define**" qui permet de définir un objet gérable par ***Nagios***, ici un hôte (**host**) :

```
define host {
        use linux-server
        host_name  serveur-linux
        address 192.168.40.1
        }
```

Cet hôte, « **serveur-linux** » ici, utilise un **modèle prédéfini** (directive « use » qui a pour valeur « **linux-server** »).
Il est **identifié** par son **nom** « **serveur-linux** » et son **adresse** « **192.168.40.1** ».

Le paramétrage est facilité par l'utilisation de **modèles** (**templates**), eux aussi déclarés par les mots clés « **define host** ». Un modèle permet de définir l'ensemble des paramètres nécessaires à l'exécution des commandes associées à ce modèle :

```
define host {
        name  linux
        use     generic-host
        register 0
        }
```

Ce modèle est **identifié par le nom « linux »** et hérite lui-même d'un modèle plus général (**generic-host**). La directive « register » mise à la valeur 0 indique qu'il s'agit d'un modèle et non d'un hôte.

Par défaut, tous les modèles font référence à un modèle générique qui teste, via une commande ***ping***, la présence d'un hôte sur le réseau. Ainsi, l'interface *web* de **Nagios** doit afficher au minimum le statut (***UP*** ou ***DOWN***) de chaque hôte déclaré dans le répertoire ***/usr/local/nagios/etc/objects***.

De nombreux modèles sont disponibles en téléchargement sous forme de « **packs** » et chaque organisation peut développer ses propres modèles.

III.3. Comment superviser les services d'un élément actif ?

La supervision d'un service se fait via une commande généralement de la forme « **check_nom-commande** » (par exemple, la commande « **check_ping** » va permettre de vérifier qu'un hôte est joignable). De même, pour vérifier une connexion **SSH**, il est possible de définir dans un « pack » le service correspondant :

```
define service {
        service_description SSH
        use     generic-service
        host_name     serveur-linux
        check_command check_ssh
        register        0
        }
```

Par conséquent, tous les hôtes utilisant ce modèle auront ce service supervisé.

Les différents états gérés par ***Nagios*** sont les suivants pour un hôte (c'est-à-dire tout élément accessible par son adresse IP) :

- ***UP*** : l'hôte répond ;
- ***DOWN*** : l'hôte ne répond pas ;
- ***UNREACHABLE*** : l'hôte est injoignable car il se trouve derrière un autre hôte qui ne répond pas ;
- ***PENDING*** : l'hôte n'est pas encore testé (au démarrage généralement).

III.4. Procédure de configuration d'un élément à superviser

Pour configurer la supervision d'un élément via le logiciel ***Nagios***, il est nécessaire de respecter les étapes suivantes :

Étape 1 : installation éventuelle du ou des modèles de configuration nécessaires dans le répertoire : ***/usr/local/nagios/etc/objects/monreso***.

Étape 2 : création du fichier de configuration (***nom_hôte.cfg***) dans le répertoire : ***/usr/local/nagios/etc/objects/monreso***.

Étape 3 : configuration du fichier créé à l'étape 2

```
define host {
        use                     nom_modèle_utilisé
        host_name               nom_hôte
        address                 adresse_IP
```

```
...
}
```

Étape 4 : définition des services

```
define service {
        service_description     nom_service
        use                     nom_modèle_utilisé
        host_name               nom_modèle
        check_command           commande
        ...
        }
```

IV - Exemples de configuration de Nagios

IV.1. Présentation

Dans le dossier **/usr/local/nagios/etc/objects/monreso**, on se propose de créer un fichier pour les groupes **hostgroups.cfg** et un fichier pour chaque machine à surveiller (on peut se limiter à un seul ficher contenant tous les hosts).

IV.2. Fichier groupes

Par exemple, **/usr/local/nagios/etc/objects/monreso/hostgroups.cfg** :

```
define hostgroup{
        hostgroup_name   linux-clients
        alias linux-clients
        }
define hostgroup{
        hostgroup_name   windows-clients
        alias windows-clients
        }
define hostgroup{
        hostgroup_name routers
        alias routers
        }
define hostgroup{
        hostgroup_name reseau40
        alias reseau40
        }
```

IV.3. Fichiers machines

Par exemple, **/usr/local/nagios/etc/objects/monreso/serveur-linux.cfg** :

```
define host {
   use                  linux-server
   host_name            serveur-linux-40
   alias                serveur-linux-40
   address              192.168.40.1
   hostgroups           linux-servers, reseau40
}
define service {
   use                  generic-service
   host_name            serveur-linux-40
   service_description  PING
   check_command        check_ping!100.0,20%!500.0,60%
```

```
}
define service {
   use                     generic-service
   host_name               serveur-linux-40
   service_description     SSH
   check_command           check_ssh
   notifications_enabled   0
}
define service {
   use                     generic-service
   host_name               serveur-linux-40
   service_description     HTTP
   check_command           check_http
   notifications_enabled   0
}
```

Par exemple, **/usr/local/nagios/etc/objects/monreso/client-linux.cfg** :

```
define host {
   use                     linux-server
   host_name               client-linux-40
   alias                   client-linux-40
   address                 192.168.40.2
   hostgroups              linux-clients, reseau40
}
define service {
   use                     generic-service
   host_name               client-linux-40
   service_description     PING
   check_command           check_ping!100.0,20%!500.0,60%
}
```

Par exemple, **/usr/local/nagios/etc/objects/monreso/client-windows.cfg** :

```
define host {
   use                     windows-server
   host_name               client-w7-40
   alias                   client-w7-40
   address                 192.168.40.3
   hostgroups              windows-clients, reseau40
}
define service {
   use                     generic-service
   host_name               client-w7-40
   service_description     PING
   check_command           check_ping!100.0,20%!500.0,60%
}
define service {
   use                     generic-service
   host_name               client-w7-40
   service_description     UPTIME
   check_command           check_snmp!-C private -o 1.3.6.1.2.1.1.3.0
}
define service {
   use                     generic-service
   host_name               client-w7-40
   service_description     CLIENTVERSION
   check_command           check_nt!CLIENTVERSION! -s snir
}
```

Par exemple, **/usr/local/nagios/etc/objects/monreso/cisco-router.cfg** :

```
define host {
   use                     linux-server
   host_name               cisco-2691
   alias                   cisco-router
   address                 192.169.0.100
   hostgroups              routers
}
define service {
   use                     generic-service
   host_name               cisco-2691
   service_description     PING
   check_command           check_ping!100.0,20%!500.0,60%
}
define service {
   use                     generic-service
   host_name               cisco-2691
   service_description     UPTIME
   check_command           check_snmp!-C private -o 1.3.6.1.2.1.1.3.0
}
```

Remarque : Pour superviser une machine **Windows**, on peut installer l'agent **NSClient++** sur le poste Windows. Il faut télécharger la dernière version stable de NSClient++ depuis http://sourceforge.net/projects/nscplus puis l'installer.

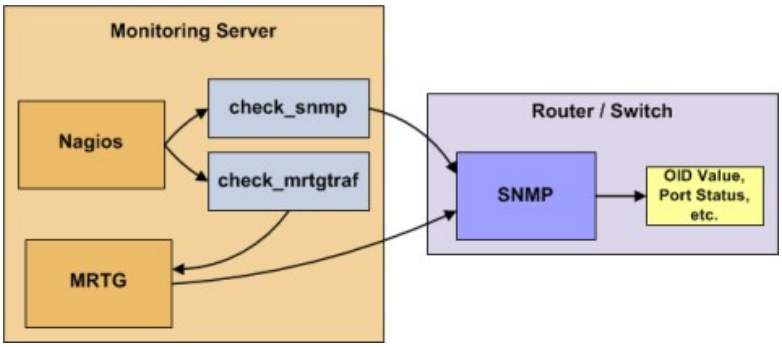
V - Les principaux Addons Nagios

V.1. Introduction

Les **addons** sont des extensions du **Nagios Core** permettant d'en étendre les fonctionnalités.

V.2. Net-SNMP

Pour surveiller un équipement via **SNMP** depuis **Nagios**, on peut utiliser la commande **check_snmp** qui dérive de l'application **Net-SNMP**.



D'une manière générale, la syntaxe de la commande **check_snmp** est la suivante :

```
check_snmp -H <Address-IP> -o <OID> [-p port]  [-P protocol]
        [-C community]    [-w warnning]    [-c critical]  [-s string]
        [-r regex] [-R regexi] [-l label] [-u units]  [-d delimiter]
        [-D output- delimiter] [-t timeout] [-e retries]  [-m miblist]
        [-L seclevel] [-U secname] [-a authproto]
        [-A authpasswd] [-X privpasswd]
```

Par exemple pour lire l'**uptime** (le temps depuis lequel le système est en service) :

/usr/local/nagios/libexec/check_snmp -H 192.168.40.3 -C private -P 1 -o .1.3.6.1.2.1.1.3.0
SNMP OK - Timeticks: (372828) 1:02:08.28 | iso.3.6.1.2.1.1.3.0=372828

Par exemple pour connaître la mémoire vive utilisée :

/usr/local/nagios/libexec/check_snmp -H 192.168.40.3 -C private -P 1 -o .1.3.6.1.4.1.9.9.48.1.1.1.5.1 -w 92000000 -c 123000000
SNMP OK - 9256592 | iso.3.6.1.4.1.9.9.48.1.1.1.5.1= 9256592;92000000;123000000

Il suffit de définir l'**hôte**, la **commande** et le **service** correspondant :

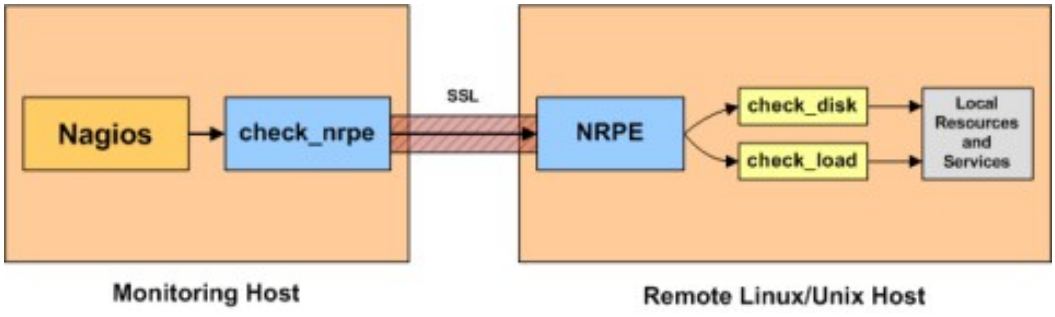
```
define host {
   use                     linux-server
   host_name               cisco-2691
   alias                   cisco-router
   address                 192.169.0.100
   hostgroups              routers
}

# Définition de la commande Check_snmp
define command {
   command_name            Check_snmp
   command_line            $USER1$/check_snmp -H $HOSTADDRESS$ -C private
-P 1 -o $ARG1$ -w $ARG2$ -c $ARG3$
}

# Appel de la commande Check_snmp
define service {
   host                    cisco-2691
   service_description     MEM_USE
   check_command           Check_snmp!.1.3.6.1.4.1.9.9.48.1.1.1.5.1!92000000!
123000000
   use generic-service
}
```

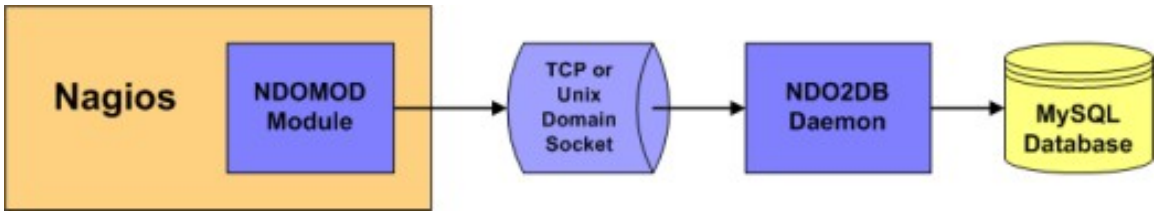
V.3. NRPE

NRPE (**N**agios **R**emote **P**lugin **E**xecutor) est un agent de supervision qui vous permet de récupérer les informations à distance. Son principe de fonctionnement est simple : il suffit d'installer le démon sur la machine distante et de l'interroger à partir du serveur **Nagios**. Il est défini comme l'agent d'interrogation de type actif car c'est le serveur **nagios** qui va interroger la machine distante. **https://support.nagios.com/kb/article.php?id=515#Debian**.



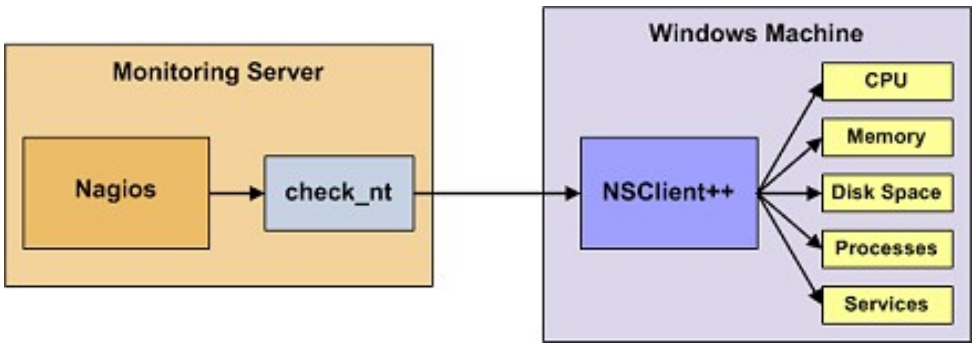
V.4. NDOUtils

NDOUtils est un addon servant à injecter les informations de Nagios en Base de données **MySQL**. Ceci permet de ne plus avoir l'ancienne gestion des archives via fichiers logs. Cet addon a permis d'avoir une plus grande ouverture sur l'exploitation des résultats de **Nagios** et de transformer l'information de la manière que l'on souhaite.



V.5. NSCLIENT++

NSCLIENT++ est un addon permettant de récupérer un nombre important d'informations à surveiller sur une machine **Windows**. Comme les plugins **NRPE** (disponible seulement sous Linux et Mac OS X), **NSClient** se base sur une architecture client/serveur. La partie cliente (nommée **check_nt**), doit être disponible sur le serveur **Nagios**. La partie serveur (**NSClient++**) est à installer sur chacune des machines **Windows** à surveiller.



V.6. PNP

PNP est un addon de métrologie (graphage des données de performances). Il permet de récupérer la partie performance de la sortie des plugins et d'injecter ces valeurs dans des bases **rrdtool** puis de les grapher via un front-end écrit en PHP.

V.7. NAGVIS

NagVis est un addon de visualisation pour Nagios qui permet de générer des vues métier de la supervision. Facile à installer, à utiliser, très intuitif avec Nagios et son système de Drag n'Drop.