

1

TP1. installation Nagios Core

▼ 1. Préparation du serveur Debian

Notre serveur Debian sera basé sur Debian Blue en version Minimal.

Installation du serveur Apache

Pour pouvoir accéder à l'interface web de gestion de Nagios nous aurons donc besoin d'un serveur Apache et de l'interpréteur PHP.

```
apt-get install apache2 php php-gd php-imap php-curl -y
```

essayer aussi le package: `php-mcrypt`

Installation des librairies PERL

Afin d'exploiter au maximum les possibilités offertes par les **Plugins** de Nagios, nous aurons besoin de quelques librairies PERL supplémentaires:

```
apt-get install libxml-libxml-perl libnet-snmp-perl libperl-dev libnumber-format-perl libconfig-inifiles-perl libdatetime-perl libnet-dns-perl -y
```

Par exemple on peut noter que sans les libraires Perl SNMP le plugin SNMP de Nagios ne sera pas fonctionnel.

Installation des librairies Graphiques

Nagios nécessite également quelques librairies graphiques

```
apt-get install libpng-dev libjpeg-dev libgd-dev -y
```

Installez les outils de compilation standards

enfin pour installer Nagios et ses plugins vous aurez besoin des outils de compilation standards et du package unzip:

En effet, Nagios est un programme compilé en C, donc le compilateur GCC et ses outils associés MAKE et AUTOCONF sont indispensables, et les sources sont téléchargées souvent dans un format compressé, donc UNZIP est nécessaire également.

```
apt-get install gcc make autoconf libc6 unzip -y
```

Installation librairie SSL

```
apt-get install libssl-dev -y
```

Création de l'environnement Nagios

La création de l'environnement Nagios englobe la création de son utilisateur, son groupe et son répertoire de travail.

En effet Nagios est un programme qui n'a pas besoin de tourner sous root.

Pour ajouter l'utilisateur Nagios sur le système, saisissez la commande suivante:

```
useradd -m -p $(openssl passwd nagios) nagios
```

Cette commande crée l'utilisateur nagios sur le système et initialise son mot de passe à nagios.

par ailleurs, il est également nécessaire de partager un groupe commun entre l'utilisateur nagios et l'utilisateur www-data (qui fait tourner le serveur Web Apache2) afin, notamment, de permettre l'administration de Nagios depuis l'interface Web.

```
groupadd nagcmd
```

```
usermod -a -G nagcmd nagios
```

```
usermod -a -G nagcmd www-data
```

1. La première commande crée un groupe `nagcmd`, qui servira à **executer des commandes Nagios depuis l'interface WEB**.
2. La seconde commande ajoute l'utilisateur `nagios` dans le groupe `nagcmd`.
3. La troisième commande ajoute l'utilisateur `www-data`, sous lequel tournent les processus du serveur Web Apache2 dans le groupe `nagcmd`. Dernière commande à passer, la **création d'un répertoire de stockage pour tous les téléchargements**.

```
mkdir /home/nagios/downloads
```

▼ 2. Téléchargement et compilation des sources

Installez les sources nagios dans le dossier `/home/nagios/downloads`

```
cd /home/nagios/downloads
```

```
wget https://sourceforge.net/projects/nagios/files/nagios-4.x/nagios-4.4.13/nagios-4.4.13.tar.gz
```

décompresser l'archive

```
tar -zxvf nagios-4.4.13.tar.gz
```

La compilation des sources Nagios passe par un script `configure`. Celui-ci permet, entre autre, de s'assurer que les éléments nécessaires sont présents sur le système, et de passer quelques paramètres au processus de compilation. Dans notre cas, nous allons indiquer deux choses: où se trouve le répertoire par défaut de configuration des sites Web et où se situe le groupe `nagcmd` que vous souhaitez configurer... le tout, avec la commande suivante:

```
cd nagios-4.4.13
```

```
./configure --with-httpd-conf=/etc/apache2/sites-enabled --with-command-group=nagcmd
```

une fois la configuration OK, faites:

```
make all
```

▼ 3. Installation de l'arborescence

Afin d'installer créer l'arborescence Nagios et d'y installer les fichiers binaires:

```
make install
```

pour vérifier que l'arborescence Nagios a été créée avec succès: `ls -lrtha /usr/local/nagios`

Installation du service Nagios

Installation du service Nagios qui se traduit par les composants nécessaires au démarrage de Nagios avec la machine.

```
make install-daemoninit
```

Installation le pipe de Nagios

Le pipe est un fichier socket pipe et ce fichier aura pour rôle de faire la liaison entre les actions des utilisateurs depuis l'interface d'administration et du noyau Nagios en cours d'exécution

```
make install-commandmode
```

Installation des fichiers de configurations de Nagios

Pour l'installation des fichiers de configuration de base de Nagios:

```
make install-config
```

Installation de l'interface Web'administration

Pour installer l'interface Web d'administration de Nagios:

```
make install-webconf
```

Cette commande dépose le fichier `nagios.conf` dans l'arborescence Apache (`/etc/apache2/sites-enabled`). Pour fonctionner correctement, l'interface d'administration de Nagios nécessite les modules `rewrite` et `cgi` d'Apache. Pour les activer:

```
a2enmod rewrite
```

```
a2enmod cgi
```

Configuration de l'accès Apache

Pour accéder à l'interface d'administration de Nagios, il est nécessaire de configurer un accès Apache **htaccess**:

```
htpasswd -cb /usr/local/nagios/etc/htpasswd.users nagiosadmin pass
```

cette commande crée le fichier htaccess dans l'arborescence du site d'administration (/usr/local/nagios/etc/htpasswd.users) et configure un premier utilisateur comme ci-dessous:

- **login : nagiosadmin**
- **password : pass**

Configuration des droits pour la configuration

La configuration Nagios s'effectuera directement avec le compte **nagios**, **pas besoin d'être root** pour cela, il faut tout de même attribuer les droits sur l'arborescence Nagios à l'utilisateur **nagios**:

```
(en root) chown -R nagios:nagcmd /usr/local/nagios
```

Redémarrez Apache

```
service apache2 restart
```

```
service nagios start
```

Pour vérifiez que le service Nagios tourne bien:

```
ps -edf | grep nagios
```

▼ 4. Vérification de l'interface d'administration web

Nous allons donc nous connecter à l'interface d'administration web, pour on va mettre l'url de notre serveur nagios du type `http://@ipduserveurNagios/nagios`.

Le site d'administration nous propose donc de nous connecter en renseignant notre login et notre password.

- **login : nagiosadmin**
- **password : pass**

2

TP2. Installez les plugins standards de Nagios

Ces plugins sont dits "standards" car ils sont développés et maintenus par l'équipe de développement Nagios, contrairement aux plugins dits "communautaires" qui sont eux mis à disposition par la communauté des utilisateurs Nagios.

▼ 0. Préparation du serveur Debian à recevoir les plugins standards de Nagios

Nagios précise quelques spécificités concernant l'utilisation de plugins particuliers. Toute l'information et les processus des plugins sont disponibles sur la [page du support nagios](#).

Vous pourrez y retrouver notamment les instructions pour les plugins spécifiques comme par exemple, la gestion d'un LDAP, d'une base de données MySQL ou encore de Samba:

Prerequisites - check_ldap

This is required for the `check_ldap` [plugin](#).

```
apt-get install -y libldap2-dev
```

Prerequisites - check_mysql check_mysql_query

This is required for the `check_mysql` and `check_mysql_query` plugins.

```
apt-get install -y libmysqlclient-dev
```

ou

```
apt-get install -y man-db
```

Prerequisites - check_disk_smb

This is required for the `check_disk_smb` [plugin](#).

```
apt-get install -y smbclient
```

Prerequisites - Other

Regroupement d'outils nécessaire pour l'installation des plugins.

```
apt-get install -y autoconf gcc libc6 libmcrypto-dev make libssl-dev wget bc  
gawk dc build-essential snmp libnet-snmp-perl gettext
```

▼ 1. Téléchargement des sources et compilation des plugins standards de Nagios

Les plugins standards de Nagios sont téléchargeable sur le site officiel Nagios.org.

Installation des sources plugins dans le répertoire `/home/nagios/downloads`:

```
cd /home/nagios/downloads
```

```
wget https://nagios-plugins.org/download/nagios-plugins-2.4.4.tar.gz
```

Décompressez l'archive dans sources:

```
tar -zxvf nagios-plugins-2.4.4.tar.gz
```

Positionnez-vous dans le nouveau répertoire créer ayant le nom de `nagios-plugins-2.4.4`

```
cd nagios-plugins-2.4.4/
```

Il est nécessaire de lancer le script `configure` pour compiler les *plugins* standards de Nagios. Ce script permet notamment de s'assurer que les éléments nécessaires sont présents sur le système, et de régler quelques paramètres.

Ici, en l'occurrence, nous indiquons qui sont, par défaut, l'utilisateur et le groupe propriétaire de ces *plugins*, en lançant le script :

```
./configure --with-nagios-user=nagios --with-nagios-group=nagcmd
```

On va maintenant faire compiler les binaires:

```
make
```

▼ 2. Installations des plugins dans l'arborescence

```
make install
```

Cette commande déploie les plugins standards dans l'arborescence Nagios.

Le répertoire par défaut des plugins standards Nagios est `/usr/local/nagios/libexec/`.

Pour vous en assurer, nous executons la commande:

```
ls -lrtha /usr/local/nagios/libexec
```

▼ 3. Vérification du fonctionnement des plugins sur Nagios Core

Verifiez la configuration de nagios:

```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

Ici cette commande va permettre de tester la syntaxe de la configuration de Nagios, donc à chaque ajout ou modification de la configuration comme ajouté des équipement ou des services il va falloir cette commande de vérification de la syntaxe avant de prendre en compte ses modifications.

L'idée ici est donc de faciliter l'administration de Nagios en créant un alias permettant d'exécuter cette longue commande avec une simple commande.

```
nano /home/nagios/.bashrc
```

Dans le fichier `.bashrc` ajouté:

```
alias testNagios="/usr/local/nagios/bin/nagios -v  
/usr/local/nagios/etc/nagios.cfg"
```

Nous allons donc ici donner les droits à l'utilisateur Nagios de redemarrer son propre service

Installer d'abord le package sudo

```
apt-get install sudo
```

Dans le fichier de configuration sudo ajoutons l'utilisateur Nagios

```
nano /etc/sudoers
```

ajoutons la ligne suivante:

```
echo "nagios ALL=NOPASSWD:/bin/systemctl restart nagios" >> /etc/sudoers
```

Encore une fois pour faciliter l'administration de Nagios nous créons un Alias permettant à l'utilisateur Nagios d'exécuter cette longue commande avec une simple commande.

```
nano /home/nagios/.bashrc
```

Dans le fichier .bashrc ajouté:

```
alias restartNagios="sudo systemctl restart nagios"
```

Nous pouvons maintenant tester le compte Nagios

Pour cela tout d'abord nous allons changer le shell par défaut de nagios passant de /bin/sh à /bin/bash.

Cette commande va pouvoir permettre de voir quelle est le shell par défaut de l'utilisateur:

```
getent passwd nagios
```

Maintenant nous allons donc changer notre shell par défaut et le mettre en bash:

```
sudo usermod -s /bin/bash nagios
```

Vous pouvez maintenant vérifier le fonctionnement de Nagios en vous retrouvant dans l'interface Web de Nagios accessible depuis son adresse ip:

http://@ipduserveurNagios/nagios et y vérifiez les alertes dans l'onglets "Alerts".

Nagios®

General
Home
Documentation

Current Status
Tactical Overview
Map (Legacy)
Hosts
Services
Host Groups
Summary
Grid
Service Groups
Summary
Grid
Problems
Services (Unhanded)
Hosts (Unhanded)
Network Outages
Quick Search:

Reports
Availability
Trends (Legacy)
Alerts
History
Summary
Histogram (Legacy)
Notifications
Event Log

System
Comments
Downtime
Process Info
Performance Info
Scheduling Queue
Configuration

Nagios® Core™
✓ Daemon running with PID 42737

Nagios® Core™
Version 4.4.13
June 01, 2023
Check for updates

Nagios XI
Easy Configuration
Advanced Reporting
Download

Nagios Log Server
Monitor and analyze
logs from anywhere
Download

Nagios Network Analyzer
Real-time netflow and
bandwidth analysis
Download

Get Started

- Start monitoring your infrastructure
- Change the look and feel of Nagios
- Extend Nagios with hundreds of addons
- Get support
- Get training
- Get certified

Quick Links

- Nagios Library (tutorials and docs)
- Nagios Labs (development blog)
- Nagios Exchange (plugins and addons)
- Nagios Support (tech support)
- Nagios.com (company)
- Nagios.org (project)

Latest News

- Nagios Update: XI 5.6.6
- Nagios Update: XI 5.6.5
- Nagios Update: XI 5.6.4
- More news...

Don't Miss...

- Monitoring Log Data with Nagios** - Nagios Log Server can handle all log data in one central location.
- Can Nagios monitor netflow?** - Yes! Nagios Network Analyzer can take in a variety of flow data. [Learn More](#)
- Nagios XI 5 Available Now!** - Easier configuration, Advanced Reporting, [Download Today!](#)

Page Four

Sans titre

3

TP3. Découverte des éléments constitutifs de Nagios

Fonctionnement détaillé de Nagios

Recensez les différents éléments constitutifs de Nagios

Les répertoires principaux de Nagios:

Pendant l'installation de Nagios nous avons pu noter que son répertoire natif était le suivant:

```
/usr/local/nagios
```

Dedans nous pouvons noter l'existence de différents éléments essentiels à la vitalité de Nagios.

Nous connaissons déjà deux éléments de cette arborescence qui sont:

- Le répertoire `bin`, qui contient le **programme principal** de Nagios (ainsi qu'un autre programme permettant d'établir quelques statistiques);
- Le répertoire `libexec`, qui contient l'ensemble des **plugins standards** Nagios compilés sur ce serveur.

▼ Le répertoire `etc` pour la configuration de Nagios

Le répertoire `etc` contient **l'ensemble de la configuration de Nagios**. A la racine de ce répertoire, vous retrouvez le fichier `htpasswd.users`, qui 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. Au démarrage, c'est ce fichier que le programme Nagios va lire en premier. Toute la configuration se trouve dans ce fichier ou dans d'autres fichiers qui y sont référencés.

Le répertoire `objects` contient **huits fichiers de configuration** avec une extension `.cfg`. Ces fichiers contiennent des **modèles de configuration** fournis directement par Nagios, nommés **TEMPLATE**. Cela permet de gagner pas mal de temps sur la configuration des sondes, nous pouvons trouver des fichiers tel que `printers.cfg` qui sont des modèles de configuration d'imprimante, `switch.cfg` qui sont des modèles pour les équipements du type switch, ...

▼ **Le répertoire `share` et `sbin` pour l'interface d'administration**

Les répertoires `share` et `sbin` contiennent les éléments utiles à l'interface d'administration de Nagios

- Le répertoire `sbin` contient les fichiers `cgi` compilés. Les fichiers `cgi` traitent de certaines des fonctionnalités de l'interface d'administration. Par exemple, le fichier `cmd.cgi` traite les actions utilisateurs, le fichier `notifications.cgi` les notifications, `summary.cgi` l'affichage condensé des objets supervisés, ect...
- Le répertoire `share`, lui contient le code des pages de l'interface Web. Ceux qui aiment développer pourront modifier ces fichiers à la main pour adapter l'interface à leurs besoins.

▼ **Le répertoire `var` pour les variables maintenues par Nagios**

Le répertoire `var` contient toutes les données variables maintenues par Nagios. Nagios, écrit dans le fichier `nagios.log` (fichier de trace du processus) dès qu'il se passe quelque chose, que ce soit un redémarrage du service, un équipement qui n'est plus joignable ou un service qui passe en alerte. Les archives des fichiers de tracers seront conservées dans le répertoire `archives`.

Les fichiers `status.dat` et `retention.dat` sont un peu particuliers:

- Le fichier `status.dat` contient **l'ensemble des données de supervision de Nagios**.

Ces données peuvent concerner le processus Nagios en tant que tel (nombres d'équipements/services supervisés, date de démarrage du programme, directives générales, etc.), mais aussi tous les équipements et services supervisés (dernier status détecté, temps de latence et durée d'exécution des sondes pour chaque équipement/service, etc.). **C'est un fichier très important: il appartient à Nagios et vous ne devez pas le modifier à la main. Ce fichier est mis à jour selon une directive inscrite dans le fichier `nagios.cfg`;**

- Le fichier `retention.dat` est construit un peu sur le même principe que `status.dat`. Cependant, ce fichier est généralement **mis à jour moins souvent**, car il sert uniquement lorsque le service Nagios redémarre. Le programme principal vient alors lire les valeurs inscrites dans ce fichier afin de positionner le statut et les données des équipements/services à superviser par défaut. Cela évite que toutes vos sondes soient remises à zéro à chaque redémarrage du service ! Le comportement de Nagios vis-à-vis de ce fichier est également défini dans `nagios.cfg`.

▼ Le répertoire `rw` pour le pipe de Nagios

Le répertoire `rw` contient notamment le **pipe** de Nagios.

Pour lister ce répertoire:

```
ls -l /usr/local/nagios/var/rw
```

Si votre service Nagios est actif, cette commande devrait renvoyer la sortie suivante:

```
nagios@debian:~$ ls -l /usr/local/nagios/var/rw/
total 0
prw-rw---- 1 nagios nagcmd 0 26 juil. 11:10 nagios.cmd
srw-rw---- 1 nagios nagcmd 0 26 juil. 11:10 nagios.qh
nagios@debian:~$
```

Le fichier `nagios.cmd` est le **"pipe FIFO"** (First In first Out) de Nagios. C'est un fichier très important. son attribut Linux, le petit "p", vous indique que ce fichier est de type "named pipe". Le principe est simple: c'est une connexion

directe vers le processus Nagios, qui va lire de manière continue les instructions passées dans ce fichier et les exécuter si elles sont compréhensibles. Les instructions passées dans le fichier sont nommées "**commandes externes**", et c'est sur ce mécanisme que s'appuie l'interface d'administration de Nagios pour discuter avec son processus.

Le fichier `nagios.qh`:

L'attribut Linux `s` indique que ce fichier est un "socket". Il est notamment utilisé par les processus fils créés par Nagios pour discuter avec le processus principal, mais peut aussi être utilisé par un développement spécifique pour s'adresser à Nagios.

Maîtrisez les principales directives du fichier de configuration de Nagios

Nous verrons ici les **directives les plus importantes** du fichier `nagios.cfg`.

log_file

La directive `log_file`, indique à Nagios le **chemin vers son fichier de traces**. Il est possible de changer le chemin par défaut à condition que Nagios ait les droits d'écriture.

cfg_file

La directive `cfg_file` permet d'indiquer à Nagios de manière spécifique le **chemin vers un fichier de configuration** à prendre en compte au démarrage.

Nous pouvons constater que le fichier `nagios.cfg` référence cinq fichiers de configuration contenus dans le répertoire `/usr/local/nagios/etc/objects`:

```
cfg_file=/usr/local/nagios/etc/objects/commands.cfg
cfg_file=/usr/local/nagios/etc/objects/contacts.cfg
cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg
cfg_file=/usr/local/nagios/etc/objects/templates.cfg
```

cfg_dir

Cette directive a le même objectif que `cfg_file`. cependant, plutôt que de référencer les fichiers de manière individuelle, `cfg_dir` référence tous les fichiers avec une extension `.cfg` contenus dans le chemin fournit en paramètre et ce, de manière récursive (en y incluant tous les répertoire et sous-répertoire).

status_file et status_update

Ces deux directives indiquent à Nagios le chemin vers le fichier `status.dat`, ainsi que l'intervalle de temps entre chaque mise à jour de ce fichier (en seconde).

resource_file

Cette directive indique **les fichiers contenant la définition des USER MACROS**. Par défaut, le fichier correspondant est `/usr/local/nagios/etc/resource.cfg` et permet de disposer de 256 variables. Celles-ci vont de `$USER1$` à `$USER256$` et permettent de stocker des valeurs "constantes" dans votre configuration Nagios.

D'ailleurs, Nagios définit votre toute première macro `$USER1$` en tant que `$USER1$=/usr/local/nagios/libexec`.

Souvenez-vous que ce répertoire contient l'ensemble des plugins standards compilés de Nagios. Vous pourrez donc adresser ce répertoire dans votre configuration directement en utilisant `$USER1$` lorsque nécessaire. Utiliser cette macro apporte des avantages tel que:

- Si vous êtes amené à changer le répertoire des plugins pour quelque raison que ce soit, vous n'aurez alors que ce fichier à changer pour positionner la nouvelle valeur de `$USER1$`. Vous pourrez ensuite migrer immédiatement tout votre configuration.

command_file, check_external_commands, log_external_commands

Voici la signification des trois directives liées au pipe FIFO de Nagios:

- la directive `command_file` indique le chemin vers le "**pipe FIFO**" de Nagios;

- la directive `check_external_commands` indique si Nagios doit écouter ce pipe;
- la directive `log_external_commands` indique si Nagios doit tracer les commandes exécutées via son pipe

Il existe des centaines d'autre directive, vous pouvez consulter la signification de toutes les autres directives existantes en consultant la [documentations en ligne](#).

4

TP4. Mise en place de la supervision d'un premier service

Afin de découvrir le fonctionnement de Nagios nous verrons ensemble les éléments qui constituent le système de supervision: *les plugins, la command, le host et le service.*

Pour cela nous mettrons en place la supervision d'un équipement et du service associé: le serveur Nagios lui-même.

Comment Nagios se supervise lui-même

Le fonctionnement de Nagios se base sur quatre éléments essentiels:

- les **plugins**. Ce sont les programmes exécutable par Nagios: binaires compilé, script Perl, script Bash, ect...
- la **command**. Cet objet appartient à Nagios et permet de définir le lancement d'un *plugin*;
- le **host**. Cet objet appartient à Nagios et représente tous les équipement à superviser. Ils possèdent généralement une adresse IP et peuvent répondre aux ping ICMP standards.
- le **service**. Cet objet appartient à Nagios et représente toutes les données à superviser sur un host. On peut citer, entre autres, les services réseaux, par exemple les ports ouverts d'un switch ou des services applicatifs, le service SSH ou HTTP d'un serveur, mais aussi la charge CPU.

Les plugins

Ce programme contient, en fin de compte, toute l'intelligence de Nagios. C'est lui qui est responsable de contacter le *host* ou le *service* à superviser, de tester son statut et de récupérer les données associées.

La liste de tous les plugins est disponible dans le répertoire `/usr/local/nagios/libexec`

La notion la plus importante à comprendre ici réside dans le fait que le *plugin* fonctionne de manière autonome et indépendante !

Exemple: **Le plugin Check_ping**

Executer le plugin `check_ping` sous le compte `nagios` (qui est le compte du service `nagios`):

```
/etc/local/nagios/libexec/check_ping
```

Cette commande produit la sortie suivante:

```
nagios@debian:/usr/local$ /usr/local/nagios/libexec/check_ping
check_ping: Impossible de décomposer les arguments
Utilisation:
check_ping -H <host_address> -w <wrta>,<wpl>% -c <crta>,<cpl>%
[-p packets] [-t timeout] [-4|-6]
nagios@debian:/usr/local$ _
```

Le plugin vous indique ici qu'il attend un minimum d'arguments, notamment l'adresse du *host* à vérifier (adresse IP ou nom si Nagios a accès à un DNS), ainsi que deux arguments:

- `w`, pour le seuil de warning, et
- `c`, pour le seuil CRITICAL.

Ces deux arguments prennent pour valeur le temps nécessaire au retour du ping et/ou le pourcentage de paquets perdus par le réseau.

Relancez alors la commande avec les arguments:

```
/etc/local/nagios/libexec/check_ping -H localhost -w 40,40% -c 60,60%
```

Le premier argument demande de vérifier que le serveur `localhost` est bien défini de la même manière que dans le fichier `/etc/hosts` du serveur. Le second

argument indique un seuil WARNING à 40ms et/ou 40% de paquet perdus. Le troisième argument indique un seuil CRITICAL à 60ms et/ou 60% de paquets perdus. Bien entendu, ces seuils ne devraient pas se déclencher avec une vérification sur localhost!

```
nagios@debian:/usr/local$ /usr/local/nagios/libexec/check_ping -H localhost -w 40,40% -c 60,60%  
PING OK - Paquets perdus = 0%, RTA = 0.11 ms|rta=0.111000ms;40.000000;60.000000;0.000000 p1=0%;40;60;0
```

Vous pouvez constater que le plugin vous envoie OK avec un certain nombre d'informations, notamment le temps d'aller-retour de la requête ainsi que le pourcentage de paquet perdus. Ces informations sont comparées aux seuils passés en paramètre afin de déterminer le statut de cette supervision. Vous comprenez ici que **ce sont bien les PLUGINS** qui, de manière indépendante, **déterminent le statut** de l'objet à superviser.

Les commandes

Au-dessus des plugins se trouvent les commandes Nagios. Deux caractéristiques les définissent:

- Leur nom. L'idée est de bien nommer ces commandes pour que leurs rôles soient intelligibles au premier coup d'oeil
- Le chemin vers le plugin à lancer avec, éventuellement, les arguments associés

Pour définir une commande qui exécuterait le plugin `check-ping` sur `localhost`, on va créer un objet `command`, dans le **cfg_file** approprié ici ça sera **commands.cfg**, on y entrera la syntaxe suivante:

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

Comme vous pouvez le constater, le nom de la commande est très explicite, et la ligne de la commande correspond parfaitement à notre exécution précédente.

Comme vue dans le TP3. nous pouvons ici utiliser la notion de **USER MACRO**. Vous avez pu constater que Nagios proposait par défaut une première macro nommée \$USER1\$, qui référençait justement le chemin vers les plugins. Vous pouvez donc transformer cette déclaration de commande:

```
define command {  
command_name check-ping-localhost  
command_line $USER1$/check_ping -H localhost -w 40,40% -c 60,60%  
}
```

Ainsi, si à l'avenir vos plugins changent de place, vous n'aurez pas à toucher à votre configuration de commande: il vous suffira, tout simplement, de modifier le fichier `ressource.cfg` !

Les Hosts

Afin de pouvoir créer un nouvel objet à superviser (équipement ou service) nous définirons l'objets dans sont **cfg_file** approprié ici ça sera `localhosts.cfg`.

L'**objet host** dispose d'un peu plus de directive que l'objet `command` donc pour cela nous nous appuyerons sur la documentation en ligne mise à disposition par l'équipe Nagios [accessible ici](#)

Vous allez devoir définir les directives minimales suivantes:

- **host_name**: c'est le nom de l'objet tel qu'il sera reconnu par Nagios. Le champ est libre, mais il faut être le plus précis possible afin d'éviter les confusions;
- **address**: l'adresse IP ou le nom pour joindre l'objet (par exemple **localhost**, **www.unsupersite.com**, etc...);
- **max_check_attempts**: Cette directive spécifique à Nagios le nombre de tentatives successives maximum renvoyant un résultat négatif avant que Nagios considère ce résultat comme fiable et passe l'objet au statut déclenchant des notifications. C'est une directive très importante sur laquelle vous devrez vous documenter sérieusement, notamment pour maîtriser les notions de statut **SOFT** et **HARD**;

- **check_command:** c'est une directive qui fait liaison entre votre commande précédemment définie et cet objet host !

```
define host {  
    host_name Nagios Server  
    address localhost  
    check_command check-ping-localhost  
    mac_check_attempts 3  
}
```

Les Services

Les services, qui sont des objets obligatoirement rattachés à un host. Pour les observer en détails, lancez dans un premier temps la commande suivantes:

```
/usr/local/nagios/libexec/check_ssh
```

Cette commande exécute un plugin Nagios, et d'après le son nom, se plugin teste probablement la réponse d'un service SSH d'un host donné.

Le seul paramètre obligatoire indiqué par ce plugin est host. Relançons donc la commande avec ce paramètre, fixé à la valeur localhost.

```
nagios@NagiosDebian:~$ /usr/local/nagios/libexec/check_ssh localhost  
SSH OK - OpenSSH_7.4p1 Debian-10+deb9u4 (protocol 2.0) | time=0,005931s;;;0,000000;10,000000
```

Le plugin renvoie "OK", le service SSH est bien disponible sur localhost.

définition du service

Tout comme pour le host, il faut, afin de superviser ce service, commencer par créer la commande associée:

```
define command{  
    command_name check-ssh-localhost  
    command_line $USER1$/check_ssh localhost
```

```
}
```

Ensuite, il faut définir un objet service. Cet objet possède ses propres directives, qui sont très bien documentées [ici](#).

Voici la liste des directives minimal pour l'objet service:

- **service_description:** c'est l'identifiant du service. Il doit être unique pour chaque host;
- **host_name:** c'est la référence à l'objet host hébergeant le service à superviser. Attention: la valeur de cette directive doit correspondre à celle de la directive **host_name** de l'objet host concerné;
- **check_command:** cette directive permet d'indiquer la commande à exécuter pour effectuer la supervision;
- **max_check_attempt:** possède la même signification que pour le host.

La définition d'un tel objet service donnerait quelque chose comme:

```
define service {  
service_description SSH sur Nagios Server  
host_name Nagios Server  
check_command check-ssh-localhost  
max_check_attempt 3  
}
```

▼ Place à la pratique !

Créez un nouveau fichier `nagios-server.cfg` dans votre répertoire de travail `/usr/local/nagios/openssl_conf`, en utilisant la commande suivante :

```
touch /usr/local/nagios/openssl_conf/nagios-server.cfg
```

Rassemblez les lignes précédemment vues dans la section précédente dans ce fichier :

```

define command {
command_name check-ping-localhost
command_line $USER1$/usr/local/nagios/libexec/check_ping -H
localhost -w 40,40% -c 60,60%
}
define command {
command_name check-ssh-localhost
command_line $USER1$/check_ssh localhost
}
define host {
host_name Nagios Server
address localhost
check_command check-ping-localhost
max_check_attempts 3
}
define service {
service_description SSH sur Nagios Server
host_name Nagios Server
check_command check-ssh-localhost
max_check_attempts 3
}

```

Enregistrez votre fichier et lancez la commande `testNagios` pour effectuer une première vérification :

```
nagios@NagiosDebian:~$ testNagios
```

Le résultat de cette commande devrait ressembler à :

```

Nagios Core 4.4.2
Copyright (c) 2009-present Nagios Core Development Team and
Community Contributors
[...]
Total Warnings: 4
Total Errors: 0

```

Things look okay - No serious problems were detected during the pre-flight check

Vous allez constater trois avertissements sur notre configuration minimale pour le service, et un avertissement sur celle du *host*. C'est normal, certaines directives ne sont pas encore définies, mais ce n'est pas bloquant ! Pour que le service prenne cela en compte, vous pouvez le relancer avec la commande `restartNagios`.

```
nagios@NagiosDebian:~$ restartNagios
nagios@NagiosDebian:~$
```

Si tout se passe bien, cette commande ne devrait rien renvoyer en sortie. Pour vérifier que votre nouvelle configuration a bien été prise en compte, rendez-vous sur l'interface d'administration de Nagios. Dans un premier temps, cliquez sur le lien « Hosts » à gauche de la page :

Current Network Status
Last Updated: Sat Nov 24 14:18:42 CET 2018
Updated every 90 seconds
Nagios® Core™ 4.4.2 - www.nagios.org
Logged in as *nagiosadmin*

Host Status Totals

Up	Down	Unreachable	Pending
2	0	0	0

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
9	0	0	0	0

Host Status Details For All Host Groups

Limit Results: 100

Host	Status	Last Check	Duration	Status Information
Nagios Server	UP	11-24-2018 14:17:39	0d 0h 1m 3s	PING OK - Paquets perdus = 0%, RTA = 0.06 ms
localhost	UP	11-24-2018 14:17:39	1d 1h 36m 3s	PING OK - Paquets perdus = 0%, RTA = 0.08 ms

Results 1 - 2 of 2 Matching Hosts

Interface « Host » de Nagios

Vous y retrouvez votre nouvel équipement supervisé « Nagios Server ». Cliquez maintenant sur le lien « Services » à gauche de la page :

Current Network Status

Last Updated: Sat Nov 24 14:21:23 CET 2018
Updated every 90 seconds
Nagios® Core™ 4.4.2 - www.nagios.org
Logged in as nagiosadmin

[View History For all hosts](#)
[View Notifications For All Hosts](#)
[View Host Status Detail For All Hosts](#)

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	0	0	0
All Problems		All Types		
0		9		

Service Status Details For All Hosts

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
Nagios Server	SSH sur Nagios Server	OK	11-24-2018 14:18:29	0d 0h 22m 54s	1/3	SSH OK - OpenSSH_7.4p1 Debian-10+deb9u4 (protocol 2.0)
localhost	Current Load	OK	11-24-2018 14:20:43	1d 1h 35m 41s	1/4	OK - Charge moyenne: 0.00, 0.00, 0.00
	Current Users	OK	11-24-2018 14:16:21	1d 1h 35m 3s	1/4	UTILISATEURS OK - 2 utilisateurs actuellement connectés sur
	HTTP	OK	11-24-2018 14:16:58	1d 1h 34m 26s	1/4	HTTP OK: HTTP/1.1 200 OK - 10975 octets en 0,003 secondes de temps de réponse
	PING	OK	11-24-2018 14:17:35	1d 1h 38m 48s	1/4	PING OK - Paquets perdus = 0%, RTA = 0.07 ms
	Root Partition	OK	11-24-2018 14:18:12	1d 1h 38m 11s	1/4	DISK OK - free space: / 4231 MB (74,76% inode=87%):
	SSH	OK	11-24-2018 14:18:50	1d 1h 37m 33s	1/4	SSH OK - OpenSSH_7.4p1 Debian-10+deb9u4 (protocol 2.0)
	Swap Usage	OK	11-24-2018 14:19:28	1d 1h 36m 56s	1/4	SWAP OK - 100% libre (2045 MB sur un total de 2045 MB)
	Total Processes	OK	11-24-2018 14:20:06	1d 1h 36m 18s	1/4	PROCS OK: 45 processus avec ETAT = RSZDT

Results 1 - 9 of 9 Matching Services

Interface « Services » de Nagios

Vous y retrouvez la supervision de votre service « SSH sur Serveur Nagios » sur le *host* « Nagios Server ».

Vous venez d'ajouter votre serveur Nagios et son service SSH en supervision. Je vous invite maintenant à étudier le contenu du fichier

```
/usr/local/nagios/libexec
```

Vous allez y découvrir la définition d'un *host* nommé `localhost`, et de huit services définis sur le même modèle que les vôtres. Vous venez de comprendre comment Nagios se supervise lui-même !

```
define host {
  use linux-server
  host_name localhost
  alias localhost
  address 127.0.0.1
}
define service {
  use local-service
  host_name localhost
  service_description PING
  check_command check_ping!100.0,20%!500.0,60%
```

```
}  
define service {  
    use local-service  
    host_name localhost  
    service_description Root Partition  
    check_command check_local_disk!20%!10%!/  
}  
[...]
```

5

TP5. Administrez Nagios via son interface

L'interface d'administration de Nagios est pratique, mais elle reste le point faible de la solution Nagios Core. "Même s'il y a des technologies récentes telles que Bootstrap et AngularJS, développer cette interface n'est pas la priorité des équipes Nagios, Nagios XI, version payante sous licence, comble parfaitement ce manque: la beauté de son interface est impressionnante, mais le prix du logiciel l'est aussi !

Menu "General"

Le menu "General" contient deux sous-menus:

- "Home", qui vous mène à la page d'accueil,
- "Documentation", qui vous amène sur la page officielle de la documentation Nagios Core.

Menu "Current Status"

Tactical Overview

Ce menu se concentre sur les données de supervision des équipements et des services. Le sous-menu "**Tactical Overview**" offre une vue globale des objets supervisés, et vous donne accès à quelques options de configuration.

Map

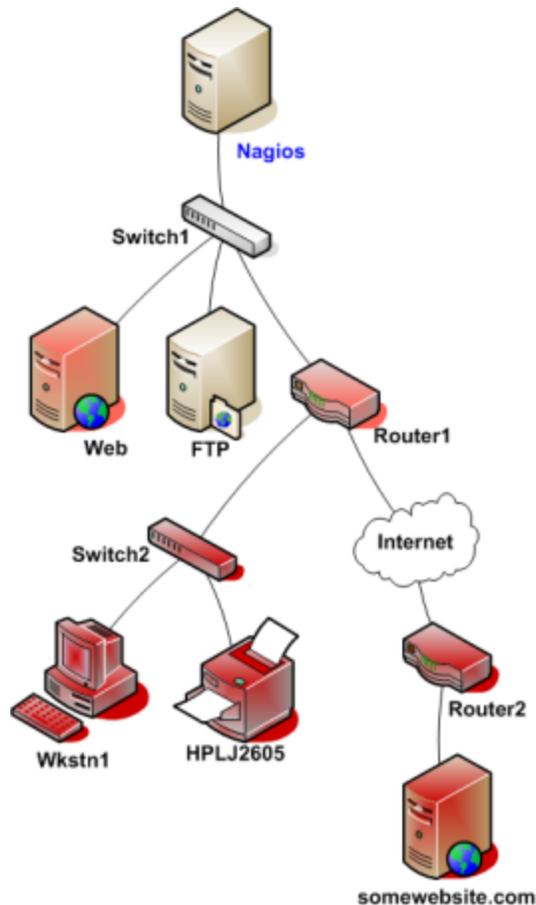
Le sous-menu "**Map**" vous permet d'obtenir une vue en fonction de la topologie définie dans les objets Nagios. Attention: elle ne reflète pas votre topologie réelle, sauf si vous la configurez intégralement dans Nagios.



Par défaut, les objets supervisés seront rattachés directement à votre serveur Nagios. C'est notamment le cas de votre host **Nagios Server**, qui se trouve au même niveau que le host **localhost** (qui est le même). La topologie Nagios peut être configurée avec la directive **parents**, au niveau des objets hosts. Il est possible d'indiquer plusieurs hosts dans cette directive, et souvent ce sont des équipements d'interconnexion comme des switches, des routeurs, des firewalls, ect...

C'est une directive **très importante** car, si Nagios détecte qu'un de ces équipements ne répond plus, il va passer tous les hosts et les services dont il est le parents en status "**UNREACHABLE**" ou "**Unknown**", status qui ne déclenchent pas de notifications.

Exemple d'architecture supervisée ([Documentation de Nagios](#)):



On observe que le Router1 est déclaré « **parent** » du Switch2 et du Router2. Lorsque le Router1 n'est plus joignable (le *plugin* « *check_ping* » renvoie **CRITICAL** par exemple), Nagios va passer en **UNREACHABLE** le statut de Switch2, de Router2, mais également de tous les équipements qui en dépendent (c'est-à-dire ici : Workstation1, HPLJ2605, et somesiteweb.com). Seuls les contacts associés à Router1 recevront une alerte

Pour la pratique, vous allez définir le *host* `localhost` en tant que parent du *host* `Nagios Server`. Pour cela, modifiez le fichier `/usr/local/nagios/openssl_conf/nagios-server.cfg` et ajoutez la directive « **parents** » dans l'objet *host* `Nagios Server` :

```

define host {
    host_name Nagios Server
    address localhost
    check_command check-ping-localhost
  }

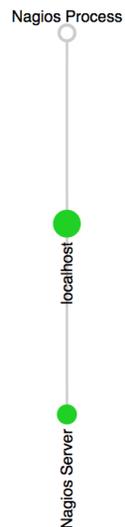
```

```
max_check_attempts 3
parents localhost
}
```

Vérifiez la configuration et relancez le service avec les commandes suivantes :

```
testNagios
restartNagios
```

Cliquez sur le sous-menu « Map » pour vérifier que cette topologie a bien été prise en compte :



Nouvelle topologie de votre réseau

Vous pouvez constater que `localhost` est maintenant le parent de `Nagios Server`.

Le menu "Hosts"

Le menu "hosts" vous propose une vue d'ensemble des objets hosts supervisés par votre serveur Nagios.

Les colonnes représentées affichent les informations suivantes:

- **Hosts:** le nom de l'objet, tel qu'il a été défini dans le fichier de configuration.
- **Status:** l'état courant du hosts:
- **Last Check:** La date de dernière exécution de la commande définie par la directive `check_command` du host.
- **Duration:** Le temps cumulé de supervision du host,
- **Status Information:** Les informations renvoyées par le plugin lors de sa dernière exécution.

La notion "**Check**": Un **Check** est l'action d'exécuter la commande définie dans la directive `check_command` d'un objet, et donc de lancer le plugin défini dans cette commande.

En cliquant sur le lien affiché sous le nom du *host*, vous obtenez un affichage détaillé de l'objet :

<p>Host Information Last Updated: Sat Nov 24 16:35:21 CET 2018 Updated every 90 seconds Nagios® Core™ 4.4.2 - www.nagios.org Logged in as nagiosadmin</p> <p>View Status Detail For This Host View Alert History For This Host View Trends For This Host View Alert Histogram For This Host View Availability Report For This Host View Notifications For This Host</p>	<p>Host localhost (localhost)</p> <p>Member of linux-servers</p> <p>127.0.0.1</p>
---	--

<p style="text-align: center;">Host State Information</p> <table border="1"> <tr><td>Host Status:</td><td>UP (for 1d 3h 52m 42s)</td></tr> <tr><td>Status Information:</td><td>PING OK - Paquets perdus = 0%, RTA = 0.08 ms</td></tr> <tr><td>Performance Data:</td><td>rta=0.079000ms;3000.000000;5000.000000;0.000000 pl=0%;80;100;0</td></tr> <tr><td>Current Attempt:</td><td>1/10 (HARD state)</td></tr> <tr><td>Last Check Time:</td><td>11-24-2018 16:32:39</td></tr> <tr><td>Check Type:</td><td>ACTIVE</td></tr> <tr><td>Check Latency / Duration:</td><td>0,000 / 4,000 seconds</td></tr> <tr><td>Next Scheduled Active Check:</td><td>11-24-2018 16:37:39</td></tr> <tr><td>Last State Change:</td><td>11-23-2018 12:42:39</td></tr> <tr><td>Last Notification:</td><td>11-23-2018 12:42:43 (notification 0)</td></tr> <tr><td>Is This Host Flapping?</td><td>NO (0,00% state change)</td></tr> <tr><td>In Scheduled Downtime?</td><td>NO</td></tr> <tr><td>Last Update:</td><td>11-24-2018 16:35:18 (0d 0h 0m 3s ago)</td></tr> </table> <table border="1"> <tr><td>Active Checks:</td><td>ENABLED</td></tr> <tr><td>Passive Checks:</td><td>ENABLED</td></tr> <tr><td>Obsessing:</td><td>ENABLED</td></tr> <tr><td>Notifications:</td><td>ENABLED</td></tr> <tr><td>Event Handler:</td><td>ENABLED</td></tr> <tr><td>Flap Detection:</td><td>ENABLED</td></tr> </table>	Host Status:	UP (for 1d 3h 52m 42s)	Status Information:	PING OK - Paquets perdus = 0%, RTA = 0.08 ms	Performance Data:	rta=0.079000ms;3000.000000;5000.000000;0.000000 pl=0%;80;100;0	Current Attempt:	1/10 (HARD state)	Last Check Time:	11-24-2018 16:32:39	Check Type:	ACTIVE	Check Latency / Duration:	0,000 / 4,000 seconds	Next Scheduled Active Check:	11-24-2018 16:37:39	Last State Change:	11-23-2018 12:42:39	Last Notification:	11-23-2018 12:42:43 (notification 0)	Is This Host Flapping?	NO (0,00% state change)	In Scheduled Downtime?	NO	Last Update:	11-24-2018 16:35:18 (0d 0h 0m 3s ago)	Active Checks:	ENABLED	Passive Checks:	ENABLED	Obsessing:	ENABLED	Notifications:	ENABLED	Event Handler:	ENABLED	Flap Detection:	ENABLED	<p style="text-align: center;">Host Commands</p> <table border="1"> <tr><td></td><td>Locate host on map</td></tr> <tr><td></td><td>Disable active checks of this host</td></tr> <tr><td></td><td>Re-schedule the next check of this host</td></tr> <tr><td></td><td>Submit passive check result for this host</td></tr> <tr><td></td><td>Stop accepting passive checks for this host</td></tr> <tr><td></td><td>Stop obsessing over this host</td></tr> <tr><td></td><td>Disable notifications for this host</td></tr> <tr><td></td><td>Send custom host notification</td></tr> <tr><td></td><td>Schedule downtime for this host</td></tr> <tr><td></td><td>Schedule downtime for all services on this host</td></tr> <tr><td></td><td>Disable notifications for all services on this host</td></tr> <tr><td></td><td>Enable notifications for all services on this host</td></tr> <tr><td></td><td>Schedule a check of all services on this host</td></tr> <tr><td></td><td>Disable checks of all services on this host</td></tr> <tr><td></td><td>Enable checks of all services on this host</td></tr> <tr><td></td><td>Disable event handler for this host</td></tr> <tr><td></td><td>Disable flap detection for this host</td></tr> <tr><td></td><td>Clear flapping state for this host</td></tr> </table>		Locate host on map		Disable active checks of this host		Re-schedule the next check of this host		Submit passive check result for this host		Stop accepting passive checks for this host		Stop obsessing over this host		Disable notifications for this host		Send custom host notification		Schedule downtime for this host		Schedule downtime for all services on this host		Disable notifications for all services on this host		Enable notifications for all services on this host		Schedule a check of all services on this host		Disable checks of all services on this host		Enable checks of all services on this host		Disable event handler for this host		Disable flap detection for this host		Clear flapping state for this host
Host Status:	UP (for 1d 3h 52m 42s)																																																																										
Status Information:	PING OK - Paquets perdus = 0%, RTA = 0.08 ms																																																																										
Performance Data:	rta=0.079000ms;3000.000000;5000.000000;0.000000 pl=0%;80;100;0																																																																										
Current Attempt:	1/10 (HARD state)																																																																										
Last Check Time:	11-24-2018 16:32:39																																																																										
Check Type:	ACTIVE																																																																										
Check Latency / Duration:	0,000 / 4,000 seconds																																																																										
Next Scheduled Active Check:	11-24-2018 16:37:39																																																																										
Last State Change:	11-23-2018 12:42:39																																																																										
Last Notification:	11-23-2018 12:42:43 (notification 0)																																																																										
Is This Host Flapping?	NO (0,00% state change)																																																																										
In Scheduled Downtime?	NO																																																																										
Last Update:	11-24-2018 16:35:18 (0d 0h 0m 3s ago)																																																																										
Active Checks:	ENABLED																																																																										
Passive Checks:	ENABLED																																																																										
Obsessing:	ENABLED																																																																										
Notifications:	ENABLED																																																																										
Event Handler:	ENABLED																																																																										
Flap Detection:	ENABLED																																																																										
	Locate host on map																																																																										
	Disable active checks of this host																																																																										
	Re-schedule the next check of this host																																																																										
	Submit passive check result for this host																																																																										
	Stop accepting passive checks for this host																																																																										
	Stop obsessing over this host																																																																										
	Disable notifications for this host																																																																										
	Send custom host notification																																																																										
	Schedule downtime for this host																																																																										
	Schedule downtime for all services on this host																																																																										
	Disable notifications for all services on this host																																																																										
	Enable notifications for all services on this host																																																																										
	Schedule a check of all services on this host																																																																										
	Disable checks of all services on this host																																																																										
	Enable checks of all services on this host																																																																										
	Disable event handler for this host																																																																										
	Disable flap detection for this host																																																																										
	Clear flapping state for this host																																																																										

Host Comments

[Add a new comment](#) [Delete all comments](#)

Entry Time	Author	Comment	Comment ID	Persistent	Type	Expires	Actions
This host has no comments associated with it							

Vue détaillée d'un objet Host

Le pavé « Host State Information » reprend les données de l'écran précédent et y ajoute quelques informations, dont :

- **Performance Data** : ces données sont renvoyées de manière normalisée par le *plugin* et peuvent être interprétées par des moteurs de rendu graphique ;
- **Next Scheduled Active Check** : indique le moment où l'objet sera à nouveau supervisé via sa commande `check_command` et le *plugin* associé, et
- **Last State Change** : indique le moment où l'objet a changé de statut pour la dernière fois.

Le pavé « **Hosts Commands** » permet d'interagir avec l'objet directement depuis l'interface d'administration via des **commandes externes** (comme nous l'avons vu dans le chapitre 3, avec le *pipe FIFO* de Nagios).

Un peu de pratique : vous allez maintenant demander à Nagios de ne pas attendre le prochain check de cet objet, mais de l'effectuer immédiatement. Pour cela, cliquez sur « **Re-schedule the next check of this host** » :

You are requesting to schedule a host check

Command Options

Host Name:

Check Time:

Force Check:

Commander un check immédiat sur l'objet

Vous pouvez décider du moment du prochain check. Par défaut, il vous sera proposé de faire un prochain check immédiatement. Il vous suffit simplement de

cliquer sur le bouton `commit` pour exécuter le check. Si tout s'est bien passé, vous devez obtenir un message de confirmation.

En cliquant sur le lien « **Done** », vous revenez à la vue détaillée de la supervision de ce *host*. Et vous pouvez constater l'exécution de votre commande externe en observant la nouvelle valeur du champ `Last Check Time`.

Pour afficher les informations, l'interface Nagios se base sur le fichier `/usr/local/nagios/var/status.dat` vu au chapitre 3. Nous avons parlé de la valeur de la directive `status_update_interval=10` dans le fichier `nagios.cfg`. Il faut comprendre que l'interface d'administration peut mettre au maximum 10 secondes pour afficher le résultat de cette commande externe. En d'autres termes, c'est le temps maximum qu'il faut à Nagios pour écrire dans le fichier. Par conséquent, si le résultat de votre commande externe ne s'affiche pas immédiatement, patientez 10 secondes et rafraîchissez la page !

Pour continuer avec ces histoires de commandes externes, répétez exactement la même opération depuis l'interface, mais en ajoutant une commande dans le terminal afin d'écouter sur le fichier de traces de Nagios en temps réel.

Dans un premier temps, exécutez la commande suivante :

```
nagios@NagiosDebian:~$ tail -f /usr/local/nagios/var/nagios.log
```

Puis, depuis le pavé de commandes de l'interface d'administration, relancez un check immédiat sur le *host* `localhost`. Vous devriez voir apparaître la ligne suivante :

```
[1543075682] EXTERNAL COMMAND: SCHEDULE_FORCED_HOST_CHECK;localhost;1543075681
```

Que vous dit cette ligne ? Vous pouvez observer, dans le premier champ entre crochets, le nombre de secondes écoulées depuis le 1er janvier 1970 à minuit : c'est le fameux *timestamp* des informaticiens. Au *timestamp* indiqué, Nagios a exécuté une commande externe nommée `schedule_force_host_check`, en prenant pour paramètre le *host* `localhost` et un nouveau *timestamp* indiquant le moment souhaité du check.

À croire que cliquer sur le lien depuis le navigateur a envoyé cette commande à Nagios (et c'est effectivement ce qui s'est passé !). Comment votre navigateur a-t-il envoyé cette commande à Nagios ? Vous l'aurez compris : en passant par le pipe FIFO de Nagios (d'où l'intérêt d'ajouter le compte `www-data` dans le groupe `nagcmd` lors de l'installation de Nagios).

En résumé, le clic sur « **Re-schedule the next check of this host** » correspond à l'écriture, dans le pipe FIFO, de la commande : `[TIMESTAMP] SCHEDULE_FORCED_HOST_CHECK;localhost;TIMESTAMP`

Vous pouvez vérifier vous-même cet état en exécutant la commande suivante (sur la même ligne !) :

```
nagios@NagiosDebian:~$ DATE=`date +%s`;echo "[${DATE}] SCHEDULE
_FORCED_HOST_CHECK;"Nagios Server";${DATE}" > /usr/local/nagios
s/var/rw/nagios.cmd
```

Dans le fichier de traces de Nagios, vérifiez l'exécution de votre commande externe avec :

```
nagios@NagiosDebian:~$ tail /usr/local/nagios/var/nagios.log

[1543073348] Warning: Service 'SSH sur Nagios Server' on host
'Nagios Server' has no check time period defined!

[1543073348] Warning: Service 'SSH sur Nagios Server' on host
'Nagios Server' has no notification time period defined!

[1543073348] Warning: Host 'Nagios Server' has no default con
tacts or contactgroups defined!

[1543073348] Successfully launched command file worker with p
id 29007

[1543075211] EXTERNAL COMMAND: SCHEDULE_FORCED_HOST_CHECK;loc
alhost;1543075116
```

```
[1543075223] EXTERNAL COMMAND: SCHEDULE_FORCED_HOST_CHECK;localhost;1543075222
```

```
[1543075300] EXTERNAL COMMAND: SCHEDULE_FORCED_HOST_CHECK;localhost;1543075299
```

```
[1543075682] EXTERNAL COMMAND: SCHEDULE_FORCED_HOST_CHECK;localhost;1543075681
```

```
[1543076216] EXTERNAL COMMAND: SCHEDULE_FORCED_HOST_CHECK;localhost;1543076216
```

```
[1543076289] EXTERNAL COMMAND: SCHEDULE_FORCED_HOST_CHECK;Nagios Server;1543076289
```

La dernière ligne correspond à votre commande pour le *host* `Nagios Server` !

Vous avez désormais compris comment fonctionnait l'interface entre l'interface Web d'administration de Nagios et le service Nagios. Toutes les commandes externes que vous allez rencontrer fonctionnent sur le même principe.

Les commandes externes Nagios sont normalisées et **disponibles ici**... il y a de quoi lire !

Le menu « Services »

Le menu « Services » vous offre une vue d'ensemble des objets *services* supervisés par votre serveur Nagios. Cet écran fonctionne sur le même principe que celui des *hosts*.

Pour voir l'affichage détaillé d'un *service*, il faut cliquer sur le lien **sous le champ « service »**, et non sous le champ « host ».

Service State Information		Service Commands	
Current Status:	OK (for 1d 4h 45m 58s)		Disable active checks of this service
Status Information:	OK - Charge moyenne: 0.00, 0.00, 0.00		Re-schedule the next check of this service
Performance Data:	load1=0.000;5.000;10.000;0; load5=0.000;4.000;6.000;0; load15=0.000;3.000;4.000;0;		Submit passive check result for this service
Current Attempt:	1/4 (HARD state)		Stop accepting passive checks for this service
Last Check Time:	11-24-2018 17:30:43		Stop obsessing over this service
Check Type:	ACTIVE		Disable notifications for this service
Check Latency / Duration:	0,000 / 0,000 seconds		Send custom service notification
Next Scheduled Check:	11-24-2018 17:35:43		Schedule downtime for this service
Last State Change:	11-23-2018 12:45:42		Disable event handler for this service
Last Notification:	11-23-2018 12:45:42 (notification 1)		Disable flap detection for this service
Is This Service Flapping?	NO (0,00% state change)		Clear flapping state for this service
In Scheduled Downtime?	NO		
Last Update:	11-24-2018 17:31:38 (0d 0h 0m 2s ago)		
Active Checks:	ENABLED		
Passive Checks:	ENABLED		
Obsessing:	ENABLED		
Notifications:	ENABLED		
Event Handler:	ENABLED		
Flap Detection:	ENABLED		

Vue détaillée d'un service Nagios

Vous aurez l'occasion de découvrir les autres sous-menus :

- « Hosts Groups », qui vous permet de visualiser la supervision des objets de type *host* par groupe. Les groupes se définissent aussi par des objets spécifiques.
- « Service Groups », qui fait la même chose pour les objets de type *services*.

Je reviendrai sur ces notions de groupe dans la troisième partie de ce cours, et enfin

- « **Problems** », qui vous permet de recenser, sur le même écran, les alertes en cours relevées par Nagios.

▼ Exploitez les outils système de l'interface d'administration

L'interface d'administration de Nagios propose également quelques fonctionnalités système à l'utilisateur.

Le menu « Reports »

Le menu « Reports » propose de construire des rapports en fonction des compteurs stockés par le moteur Nagios.

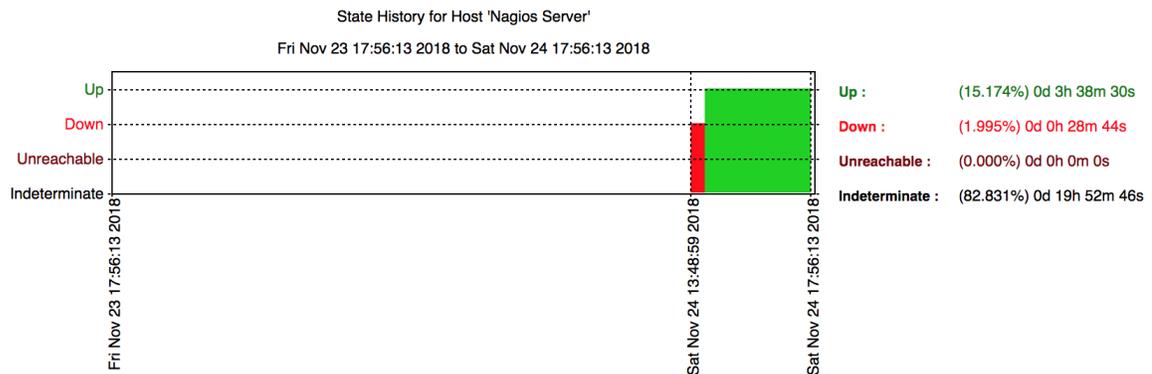
Le sous-menu « **Availability** » permet d'établir des rapports montrant les pourcentages de temps sur les états pour les objets de type *hosts* ou *services* :

Service State Breakdowns:

Host	Service	% Time OK	% Time Warning	% Time Unknown	% Time Critical	% Time Undetermined
Nagios Server	SSH sur Nagios Server	0,000% (0,000%)	0,000% (0,000%)	0,000% (0,000%)	0,000% (0,000%)	100,000%
localhost	Current Load	100,000% (100,000%)	0,000% (0,000%)	0,000% (0,000%)	0,000% (0,000%)	0,000%
	Current Users	100,000% (100,000%)	0,000% (0,000%)	0,000% (0,000%)	0,000% (0,000%)	0,000%
	HTTP	100,000% (100,000%)	0,000% (0,000%)	0,000% (0,000%)	0,000% (0,000%)	0,000%
	PING	100,000% (100,000%)	0,000% (0,000%)	0,000% (0,000%)	0,000% (0,000%)	0,000%
	Root Partition	100,000% (100,000%)	0,000% (0,000%)	0,000% (0,000%)	0,000% (0,000%)	0,000%
	SSH	100,000% (100,000%)	0,000% (0,000%)	0,000% (0,000%)	0,000% (0,000%)	0,000%
	Swap Usage	100,000% (100,000%)	0,000% (0,000%)	0,000% (0,000%)	0,000% (0,000%)	0,000%
	Total Processes	100,000% (100,000%)	0,000% (0,000%)	0,000% (0,000%)	0,000% (0,000%)	0,000%
Average		88,889% (88,889%)	0,000% (0,000%)	0,000% (0,000%)	0,000% (0,000%)	11,111%

Service state breakdowns

Le sous-menu « **Trends** » permet de construire des rapports concernant l'historique d'évolution de l'état d'un objet *host* ou *service*.



Menu « Host Trends »

Le sous-menu « **Alerts** » affiche un condensé horodaté de tous les évènements survenus aux objets *services* et *hosts* supervisé.

novembre 24, 2018 16:00

-  [11-24-2018 16:29:08] Nagios 4.4.2 starting... (PID=29001)
-  [11-24-2018 16:29:08] Caught SIGTERM, shutting down...
-  [11-24-2018 16:29:08] Caught SIGTERM, shutting down...
-  [11-24-2018 16:29:08] Caught SIGTERM, shutting down...

novembre 24, 2018 14:00

-  [11-24-2018 14:17:43] HOST ALERT: Nagios Server;UP;HARD;1;PING OK - Paquets perdus = 0%, RTA = 0.06 ms
-  [11-24-2018 14:17:31] Nagios 4.4.2 starting... (PID=27846)
-  [11-24-2018 14:17:31] Caught SIGTERM, shutting down...
-  [11-24-2018 14:11:18] Nagios 4.4.2 starting... (PID=27738)
-  [11-24-2018 14:11:18] Caught SIGTERM, shutting down...
-  [11-24-2018 14:11:18] Caught SIGTERM, shutting down...
-  [11-24-2018 14:11:18] Caught SIGTERM, shutting down...

novembre 24, 2018 13:00

-  [11-24-2018 13:58:03] Nagios 4.4.2 starting... (PID=27576)
-  [11-24-2018 13:58:03] Caught SIGTERM, shutting down...
-  [11-24-2018 13:58:03] Caught SIGTERM, shutting down...
-  [11-24-2018 13:58:03] Caught SIGTERM, shutting down...
-  [11-24-2018 13:58:03] Caught SIGTERM, shutting down...
-  [11-24-2018 13:50:59] HOST ALERT: Nagios Server;DOWN;HARD;3;(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/usr/local/nagios/libexec/check_ping, ...) failed. errno is 2: No
-  [11-24-2018 13:48:59] HOST ALERT: Nagios Server;DOWN;SOFT;1;(No output on stdout) stderr: execvp(/usr/local/nagios/libexec/usr/local/nagios/libexec/check_ping, ...) failed. errno is 2:
-  [11-24-2018 13:47:56] Nagios 4.4.2 starting... (PID=27454)
-  [11-24-2018 13:47:56] Caught SIGTERM, shutting down...
-  [11-24-2018 13:47:56] Caught SIGTERM, shutting down...
-  [11-24-2018 13:47:56] Caught SIGTERM, shutting down...
-  [11-24-2018 13:47:56] Caught SIGTERM, shutting down...
-  [11-24-2018 13:33:47] Nagios 4.4.2 starting... (PID=27262)
-  [11-24-2018 13:33:47] Caught SIGTERM, shutting down...
-  [11-24-2018 13:33:47] Caught SIGTERM, shutting down...

Menu « Alerts »

Le sous-menu « **Notifications** » affiche un résumé des différentes notifications générées par Nagios.

Enfin, le sous-menu « Event Log » affiche, tout simplement, tout le contenu du fichier de traces de Nagios `/usr/local/nagios/var/nagios.log` :

Current Event Log

Last Updated: Sat Nov 24 18:02:57 CET 2018
Nagios® Core™ 4.4.2 - www.nagios.org
Logged in as nagiosadmin



Log File Navigation

Sat Nov 24 00:00:00 CET 2018
to
Present..

Older Entries First:

File: /usr/local/nagios/var/nagios.log

novembre 24, 2018 17:00

-  [11-24-2018 17:29:08] Auto-save of retention data completed successfully.
-  [11-24-2018 17:18:09] EXTERNAL COMMAND: SCHEDULE_FORCED_HOST_CHECK;Nagios Server;1543076289
-  [11-24-2018 17:16:56] EXTERNAL COMMAND: SCHEDULE_FORCED_HOST_CHECK;localhost;1543076216
-  [11-24-2018 17:08:02] EXTERNAL COMMAND: SCHEDULE_FORCED_HOST_CHECK;localhost;1543075681
-  [11-24-2018 17:01:40] EXTERNAL COMMAND: SCHEDULE_FORCED_HOST_CHECK;localhost;1543075299
-  [11-24-2018 17:00:23] EXTERNAL COMMAND: SCHEDULE_FORCED_HOST_CHECK;localhost;1543075222
-  [11-24-2018 17:00:11] EXTERNAL COMMAND: SCHEDULE_FORCED_HOST_CHECK;localhost;1543075116

novembre 24, 2018 16:00

-  [11-24-2018 16:29:08] Successfully launched command file worker with pid 29007
-  [11-24-2018 16:29:08] Warning: Host 'Nagios Server' has no default contacts or contactgroups defined!
-  [11-24-2018 16:29:08] Warning: Service 'SSH sur Nagios Server' on host 'Nagios Server' has no notification time period defined!
-  [11-24-2018 16:29:08] Warning: Service 'SSH sur Nagios Server' on host 'Nagios Server' has no check time period defined!
-  [11-24-2018 16:29:08] Warning: Service 'SSH sur Nagios Server' on host 'Nagios Server' has no default contacts or contactgroups defined!
-  [11-24-2018 16:29:08] wproc: Registry request: name=Core Worker 29005;pid=29005
-  [11-24-2018 16:29:08] wproc: Registry request: name=Core Worker 29003;pid=29003
-  [11-24-2018 16:29:08] wproc: Registry request: name=Core Worker 29006;pid=29006
-  [11-24-2018 16:29:08] wproc: Registry request: name=Core Worker 29004;pid=29004
-  [11-24-2018 16:29:08] wproc: Successfully registered manager as @wproc with query handler
-  [11-24-2018 16:29:08] qh: help for the query handler registered
-  [11-24-2018 16:29:08] qh: echo service query handler registered
-  [11-24-2018 16:29:08] qh: core query handler registered
-  [11-24-2018 16:29:08] qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
-  [11-24-2018 16:29:08] LOG VERSION: 2.0
-  [11-24-2018 16:29:08] Local time is Sat Nov 24 16:29:08 CET 2018
-  [11-24-2018 16:29:08] Nagios 4.4.2 starting... (PID=29001)
-  [11-24-2018 16:29:08] Successfully shutdown... (PID=27846)
-  [11-24-2018 16:29:08] Caught SIGTERM, shutting down...
-  [11-24-2018 16:29:08] Caught SIGTERM, shutting down...
-  [11-24-2018 16:29:08] Caught SIGTERM, shutting down...
-  [11-24-2018 16:17:31] Auto-save of retention data completed successfully.

novembre 24, 2018 15:00

Menu « Event Log »

Le menu « System »

Le menu « System » est un menu important qui fournit beaucoup d'informations sur le comportement du serveur Nagios.

Le sous-menu « **Comments** » recense tous les commentaires des utilisateurs de l'interface d'administration concernant les événements survenus aux objets supervisés.

Le lien « **Downtime** » offre la possibilité de définir une période de temps pendant laquelle Nagios ne tiendra pas compte des résultats des *plugins* pour les notifications normales, seules les notifications indiquant le départ et l'arrêt de ces périodes seront envoyées.

You are requesting to schedule downtime for a particular host

Command Options

Host Name:

Author (Your Name):

Comment:

Triggered By:

Start Time:

End Time:

Type:

If Flexible, Duration: Hours Minutes

Child Hosts:

Fonction « Downtime »

Le sous-menu « **Process info** » permet d'obtenir des statistiques sur le fonctionnement de Nagios :

Process Information		Process Commands	
Program Version:	4.4.2	 Shutdown the Nagios process	
Program Start Time:	11-24-2018 16:29:08	 Restart the Nagios process	
Total Running Time:	0d 1h 48m 45s	 Disable notifications	
Last Log File Rotation:	N/A	 Stop executing service checks	
Nagios PID	29001	 Stop accepting passive service checks	
Notifications Enabled?	YES	 Stop executing host checks	
Service Checks Being Executed?	YES	 Stop accepting passive host checks	
Passive Service Checks Being Accepted?	YES	 Disable event handlers	
Host Checks Being Executed?	YES	 Start obsessing over services	
Passive Host Checks Being Accepted?	YES	 Start obsessing over hosts	
Event Handlers Enabled?	Yes	 Disable flap detection	
Obsessing Over Services?	No	 Enable performance data	
Obsessing Over Hosts?	No		
Flap Detection Enabled?	Yes		
Performance Data Being Processed?	No		

Menu « Process Info »

Vous constaterez que cette page contient un panel de « **Process Commands** », qui fonctionne exactement comme les « **Host Commands** » et les « **Service Commands** ». En d'autres termes, un clic sur ces liens se traduit par une commande externe envoyée sur le pipe FIFO de Nagios.

Notez qu'il est possible, entre autres, de :

- arrêter ou redémarrer le service Nagios ;
- stopper toutes les sondes pour les objets de type *host* ;
- stopper toutes les sondes pour les objets de type *services*. Le sous-menu « **Performance Info** » affiche les informations de performance :

Program-Wide Performance Information

	Time Frame	Services Checked	Metric	Min.	Max.	Average
Services Actively Checked:	<= 1 minute:	2 (22,2%)	Check Execution Time:	0,00 sec	4,00 sec	0,444 sec
	<= 5 minutes:	9 (100,0%)	Check Latency:	0,00 sec	0,00 sec	0,000 sec
	<= 15 minutes:	9 (100,0%)	Percent State Change:	0,00%	0,00%	0,00%
	<= 1 hour:	9 (100,0%)				
	Since program start:	9 (100,0%)				
Services Passively Checked:	<= 1 minute:	0 (0,0%)				
	<= 5 minutes:	0 (0,0%)	Percent State Change:	0,00%	0,00%	0,00%
	<= 15 minutes:	0 (0,0%)				
	<= 1 hour:	0 (0,0%)				
	Since program start:	0 (0,0%)				
Hosts Actively Checked:	<= 1 minute:	1 (50,0%)	Check Execution Time:	4,00 sec	4,00 sec	4,000 sec
	<= 5 minutes:	2 (100,0%)	Check Latency:	0,00 sec	0,00 sec	0,000 sec
	<= 15 minutes:	2 (100,0%)	Percent State Change:	0,00%	0,00%	0,00%
	<= 1 hour:	2 (100,0%)				
	Since program start:	2 (100,0%)				
Hosts Passively Checked:	<= 1 minute:	0 (0,0%)				
	<= 5 minutes:	0 (0,0%)	Percent State Change:	0,00%	0,00%	0,00%
	<= 15 minutes:	0 (0,0%)				
	<= 1 hour:	0 (0,0%)				
	Since program start:	0 (0,0%)				
Check Statistics:	Type	Last 1 Min	Last 5 Min	Last 15 Min		
	Active Scheduled Host Checks	1	2	6		
	Active On-Demand Host Checks	0	0	0		
	Parallel Host Checks	1	2	6		
	Serial Host Checks	0	0	0		
	Cached Host Checks	0	0	0		
	Passive Host Checks	0	0	0		
	Active Scheduled Service Checks	2	9	27		
	Active On-Demand Service Checks	0	0	0		
	Cached Service Checks	0	0	0		
	Passive Service Checks	0	0	0		
	External Commands	0	0	0		
Buffer Usage:	Type	In Use	Max Used	Total Available		
	External Commands	0	0	0		

Menu « Nagios Performance »

Cet écran est très important car il donne des indicateurs concernant « l'état de santé » du serveur Nagios, parmi lesquels :

- le nombre de checks actifs pour les *hosts* par minute ;
- le nombre de checks actifs pour les *services* par minute ;
- le temps d'exécution maximum pour les checks *hosts* et *services*.

Vous pouvez constater que, sur cette page, le check d'un *host* prend 4 secondes ! C'est beaucoup étant donné que, pour l'instant, vous surveillez uniquement le serveur Nagios.

Comment faire pour identifier le check du *host* qui prend autant de temps ?

C'est simple ! Lancez la commande suivante :

```
nagios@NagiosDebian:~$ grep -n check_execution_time /usr/local/nagios/var/status.dat
```

Observez le résultat de cette commande :

```
68: check_execution_time=4.094
123: check_execution_time=4.103
179: check_execution_time=0.007
236: check_execution_time=0.002
293: check_execution_time=0.001
350: check_execution_time=0.001
407: check_execution_time=4.095
464: check_execution_time=0.001
521: check_execution_time=0.005
578: check_execution_time=0.001
635: check_execution_time=0.008
```

Vous pouvez constater que trois sondes mettent plus de 4 secondes à s'exécuter. Consultez désormais ce fichier à partir du premier numéro de ligne renvoyé par la commande (ici, 68) :

```
56 hoststatus {
57 host_name=Nagios Server
58 modified_attributes=0
59 check_command=check-ping-localhost
60 check_period=
61 notification_period=
62 importance=0
63 check_interval=5.000000
64 retry_interval=1.000000
65 event_handler=
66 has_been_checked=1
```

```
67 should_be_scheduled=1
68 check_execution_time=4.094
```

Vous constatez alors que la sonde de votre équipement « Nagios Server » met 4 secondes, alors qu'elle est exécutée à partir de ce même équipement !

Pour vérifier ce temps d'exécution, relancez à la main le *plugin* `check_ping` que vous avez configuré dans le fichier `/usr/local/nagios/openssl_conf/nagios-server.cfg` et ajoutez devant la commande `time` permettant de relever le temps d'exécution telle que :

```
nagios@NagiosDebian:~$ time /usr/local/nagios/libexec/check
_ping -H localhost -w 40,40% -c 60,60%

PING OK - Paquets perdus = 0%, RTA = 0.04 ms|rta=0.041000m
s;40.000000;60.000000;0.000000 pl=0%;40;60;0

real 0m4,104s

user 0m0,000s

sys 0m0,000s
```

Effectivement, le *ping* du localhost prend 4 secondes !

Relancez le *plugin* avec son option `-help` pour comprendre ce phénomène et intéressez-vous notamment à l'option `-p` :

```
nagios@NagiosDebian:~$ /usr/local/nagios/libexec/check_ping
-help
```

Ci-dessous, le texte décrivant l'option `-p` :

```
-p, --packets=INTEGER
```

nombre de paquets ICMP à envoyer (Défaut: 5)

Et voilà ! La sonde met 4 secondes car elle envoie cinq paquets ! Un check sur la même machine ne sera pas forcément probant. On pourrait descendre le nombre de paquets à 1, mais dans ce cas, les seuils concernant le pourcentage de paquets perdus ne seraient plus probants. Encore une fois, pour un check local, ce n'est pas grave.

Relancez la commande :

```
nagios@NagiosDebian:~$ time /usr/local/nagios/libexec/check_ping -H localhost -w 40,40% -c 60,60% -p 1
```

```
PING OK - Paquets perdus = 0%, RTA = 0.03 ms|rta=0.027000ms;40.000000;60.000000;0.000000 pl=0%;40;60;0
```

```
real 0m0,004s
```

```
user 0m0,000s
```

```
sys 0m0,000s
```

Bingo, la sonde est bien plus rapide !

À retenir : consultez régulièrement le sous-menu « Performance Info » et assurez-vous que les valeurs importantes sont maîtrisées !

Attention aux options par défaut des *plugins* lors de l'exécution des sondes !

Il reste un dernier sous-menu, nommé « **Configuration** », que je vais volontairement omettre de détailler ici. Pour comprendre son fonctionnement, rendez-vous dans la partie 3 où l'on mettra en place une sonde HTTP.

Vous aurez remarqué qu'il n'est pas possible d'ajouter un objet à superviser directement depuis l'interface d'administration de Nagios Core. C'est le grand défaut de l'interface, que la communauté a essayé de combler avec des interfaces spécifiques, qui ne sont plus vraiment maintenues à ce jour. Je pense notamment à NCONF, qui a connu sa période de succès, ou encore

à Centreon, qui s'est bien implanté sur le marché français grâce à une interface de configuration complète !

Quelques outils offrent une interface complète, comme Eyes Of Network, que l'on peut trouver dans quelques institutions publiques françaises. Eyes of Network (produit sous licence GPL2), intègre des outils intéressants, comme CACTI.

Mais, comme je l'indiquais en introduction de ce chapitre, la nouvelle interface Web d'administration de Nagios XI est sa grande. À mon sens, il s'agit de l'interface de configuration la plus aboutie. Mais c'est un investissement...