
# Faculté des Sciences et Techniques

# La supervision avec NAGIOS

Par **Elie MABO** et **Amadou NIANG**
Etudiants en Master Informatique,
**Option** Sécurité des Systèmes Informatiques

**Janvier 2009**

# Sommaire

## C'est quoi Nagios?

Nagios est un logiciel libre de surveillance (Monitoring) des réseaux et systèmes, très connu dans le monde de l'entreprise et des professionnels réseaux. Il permet de surveiller les hôtes et services spécifiés dans son fichier de configuration, et d'alerter les administrateurs systèmes et réseaux en cas d'évènemenent (*Mauvais ou Bon*). Nagios permet la supervision **active** et **passive**.

## Un peu d'histoire

Anciennement appelé NetSaint, Nagios à l'origine était destiné uniquement pour les systèmes Linux, mais actuellement, elle peut se déployer sur n'importe quel système Unix.

## Quelques fonctionnalités de Nagios

- Surveillance des services réseaux tels que: SMTP, HTTP, FTP, SSH, etc.
- Surveillance des ressources machines telles que: Charge de processeur, Utilisation de l'espace disque, Utilisation de la mémoire, etc.
- Rotation automatique des fichiers journaux
- Interface Web optionnelle permettant de visualiser l'état actuelle du réseau, les notifications et les fichiers journaux
- Conception des simples greffons (plugins) permettant aux utilisateurs de développer leurs propres vérificateurs de services;
- Notification par mail ou sms lorsqu'un problème survient sur un service ou une machine;
- Support pour l'implémentaton d'un système de surveillance redondant;
- Etc...

## Version et Licence

Au moment de la rédaction de ce document, Nagios est actuellement à sa version 3.06. il est placé sous la Licence **GNU GPL** (General Public License) version 2. Ce qui donne la permission légale de le copier, le distribuer et/ou de le modifier sous certaines conditions. Pour plus de détails, il faut lire le fichier LICENSE inclu dans la distribution de Nagios.

La nouvelle version corrige les bugs présents dans les versions 3.0.5 et antérieures: Voici quelques unes des ces corrections:

- Correction de la soumission de commandes externes via CGI
- Correction du groupe Apache dans les spécifications pour les RPMs
- Meilleure prise en charge de l'erreur d'écriture dans les fichiers de retention et status résultant d'un disque plein
- Correction des alertes audio dans les CGIs
- Ajout du support des certificats d'authentification x509 dans les CGIs
- Etc.

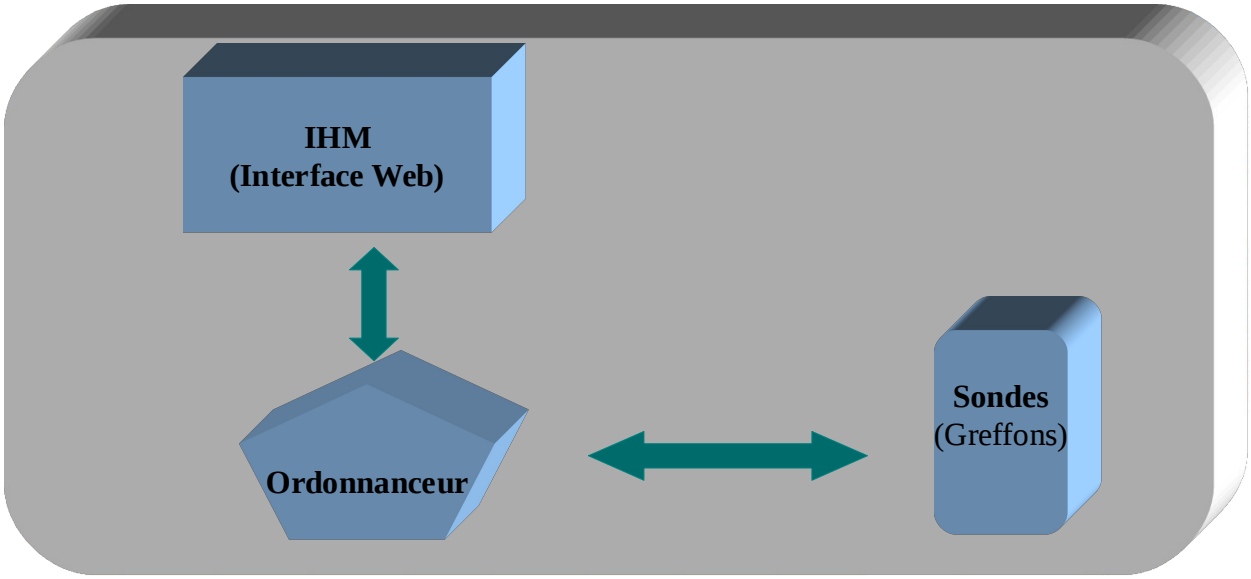
## Concepts et principe de fonctionnement de Nagios

Nagios ne possède aucun mécanisme interne pour surveiller le statut des équipements et des applications. Il repose sur des programmes externes appelés greffons (plugins). Nagios peut être assimilé à un planificateur de tâches. Il exécute un greffon à intervalle régulier lorsqu'un service ou un host doit être surveiller.

## Architecture de Nagios

Nagios peut être décomposé en trois parties:

- **Un ordonnanceur**, chargé de contrôler quand et dans quel ordre les contrôles des services sont effectués.
- Une **interface graphique** qui affiche de manière claire et concise l'état des services surveillés.
- **Des greffons**




## C'est quoi un greffon (plugin)?

Un greffon est un programme exécutable ou script (*perl, shell, etc.*) capable de fournir au moteur:

- un code de retour
  - => 0 = tout va bien (OK)
  - => 1 = avertissement (WARNING)
  - => 2 = alerte (CRITICAL)
  - => 3 = inconnu (UNKNOWN)
- un court message descriptif

En option, un greffon peut retourner des informations de performance permettant à Nagios de les interpréter pour tracer des graphiques.

## Exécution des greffons

Les greffons peuvent fonctionner localement (*directement sur la machine supervisée*) ou à distance (au travers du réseau). Pour l'exécution à distance des greffons, il exixte plusieurs possibilités:

- Par le biais d'autres serveurs de supervision Nagios distant. Cette méthode est utilisée dans le cadre de la supervision distribuée.
- Par les agents d'exécution de tests tels que: **NRPE, NSCA, check_by_ssh, NSClient**, etc...

## MISE EN PLACE DE NAGIOS

La mise en place de Nagios passe par la récupération des sources sur le site “SourceForge”, l'installation et la configuration.

## Récupération des sources

Pour installer Nagios, nous aurons besoin de deux archives: Nagios (**nagios-3.0.4.tar.gz**) et ses greffons de base (**nagios-plugins-1.4.13.tar.gz**). Ces archives sont disponibles en téléchargement sur le site officiel de Nagios (http://www.nagios.org) ou sur certains sites mirroirs. Vous trouverez les différentes versions en fonction de votre système d'exploitation. Nous avons fait des test avec le système Linux Fedora dans sa version 9.

## Installation

### Pré-requis

Le seul pré-requis pour le déploiement de Nagios est une machine exécutant Linux comme OS (ou une variante UNIX) avec un compilateur C installé.
Bien évidemment, TCP/IP doit être configuré. Il n' y a aucune contrainte avec l'utilisation des CGIs fournies avec Nagios. Mais si on souhaite les utiliser, il faut absolument installer les outils suivants:

- Serveur Web (de préférence apache)
- Librairies “gd” en version 1.6.3 ou supérieure (nécessaire pour l'utilisation des CGI “stausmap” et “trends”

## Voici les étapes d'installation

– Création d'un nouvel utilisateur nommé “nagios”

```
[root@emabolaptop emabo]# useradd -m nagios
[root@emabolaptop emabo]# passwd nagios
```

– Création d'un nouveau groupe pour les commandes externes

```
[root@emabolaptop emabo]#groupadd nagcmd
```

– Affectation des utilisateurs “nagios” et “apache” au nouveau groupe crée

```
[root@emabolaptop emabo]#usermod -G nagcmd nagios
[root@emabolaptop emabo]#usermod -G nagcmd apache
```

– Exécution du script de configuration de nagios (*cela suppose que vous avez déjà téléchargé les sources de nagios, désarchivé et que vous êtes dans le répertoire où se trouve les fichiers d'installation*)

```
[root@emabolaptop nagios]#./configure  --with-command-group=nagcmd
```

```
*** Configuration summary for nagios 3.0.2 05-19-2008 ***:

General Options:
-------------------------
        Nagios executable:  nagios
        Nagios user/group:  nagios,nagios
       Command user/group:  nagios,nagcmd
           Embedded Perl:  no
            Event Broker:  yes
        Install ${prefix}:  /usr/local/nagios
              Lock file:  ${prefix}/var/nagios.lock
  Check result directory:  ${prefix}/var/spool/checkresults
         Init directory:  /etc/rc.d/init.d
 Apache conf.d directory:  /etc/httpd/conf.d
            Mail program:  /bin/mail
                 Host OS:  linux-gnu

Web Interface Options:
------------------------
                HTML URL:  http://localhost/nagios/
```

```
CGI URL:  http://localhost/nagios/cgi-bin/
Traceroute (used by WAP):  /bin/traceroute
```

– Compilation du code source

```
[root@emabolaptop nagios]#make all
```

– Installation des binaires, du script d'initialisation, des fichiers de configuration et l'ensemble des permissions sur le répertoire des commandes externes

```
[root@emabolaptop nagios]#make install
[root@emabolaptop nagios]#make install-init
[root@emabolaptop nagios]#make install-config
[root@emabolaptop nagios]#make install-commandmode
```

– Personnalisation des fichiers de configuration

Les différents fichiers de configuration installés se trouvent dans le répertoire "`/usr/local/nagios/etc/`". Il est possible avant le démarrage de Nagios, de personnaliser les paramètres contenus dans certains fichiers. Dans notre cas, pour un début, nous avons juste édité le fichier "**contact.cfg**" et modifié la valeur de la variable "**email**" dans ce fichier.

– Configuration de l'interface Web et création d'un compte administrateur d'accès à cet interface Web

```
[root@emabolaptop nagios]#make install-webconfig
[root@emabolaptop nagios]#htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

– Redémarrage du serveur Web Apache

```
[root@emabolaptop nagios]#service httpd restart
```

– Compilation et installation des greffons nagios (*cela suppose que vous avez déjà téléchargé les sources de "nagios-plugins", désarchivé et que vous êtes dans le répertoire où se trouve les fichiers d'installation*)

```
[root@emabolaptop  nagios-plugins-1.4.13]#./configure --with-nagios-user=nagios --with-nagios-group=nagios
root@emabolaptop  nagios-plugins-1.4.13]#make
root@emabolaptop  nagios-plugins-1.4.13]#make install
```

– Démarrage de Nagios

Avant de démarrer nagios, il faut vérifier son fichier de configuration avec la commande suiivante:

```
root@emabolaptop  nagios]#/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

En suite, si tout va bien,

```
[root@emabolaptop  nagios]#service nagios start
```

Une fois l'installation terminée, Il est possible d'accéder à l'interface Web de supervision de Nagios en tapant dans la barre d'adresse de votre navigateur: "**http://localhost/nagios/**"

**Note:** Il peut y avoir un problème d'accès à cette interface, si un firewall est installé sur le serveur. Dans notre cas (Fedora 9), nous avons de modifier le mode de fonctionnement du firewall "SELinux". Nous sommes passé du mode "**Enforcing**" au mode "**Permissive**". Cette modification se fait dans le fichier de configuration "`/etc/selinux/config`". N'oubliez par de redémarrer le service firewall.

## Accès à l'interface Web d'administration de Nagios

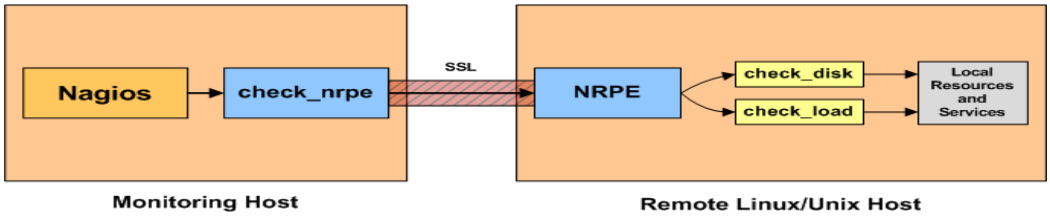
Pour accéder à l'interface Web d'administration de Nagios, il suffit de taper le lien suivant dans la barre d'adresse de votre navigateur: "`http://nom_de_votre_serveur/nagios`". Dans notre cas nous avons taper "`http://localhost/nagios`". Tout dépend du répertoire que vous avez précisé lors de l'installation de Nagios.
Vous devriez en suite entrer un nom de connexion ("*nagiosadmin" dans notre cas*) et un mot de passe pour avoir l'accès à cette interface Web. Ce compte est stocké dans le fichier "**htpasswd.users**" qui se trouve dans le répertoire des fichiers de configuration de Nagios. Dans notre cas, il se trouve dans "`/usr/local/nagios/etc/`".


**Note:** Vous pouvez utiliser la commande "`htpasswd`" pour gérer (*création, modification, suppression*) les comptes ayant le droit d'accéder à cette interface Web. Un "**man**" sur cette commande vous donne les options pouvant être utilisées avec cette commande.

Une fois l'installation de Nagios terminée, il faut éditer un certains nombre de fichiers de configuration et modifier certains paramètres afin que la supervision des machines distantes soient opérationnelle. Cependant, les fichiers de configuration par défaut de Nagios permettent tout de même de superviser quelques resources (disques dur, CPU, etc...) de la machine locale (Machine sur laquelle Nagios a été installé).

## Supervision des machines Linux

La supervision des machines Linux se fait grâce à l'agent **NRPE** qui doit être installé sur la machine distante à superviser. Le schéma suivant présente les différents composants qui doivent être mis en place et leur interaction pour que la supervision soit opérationnelle.




**Source image:** Manuel officiel de nagios

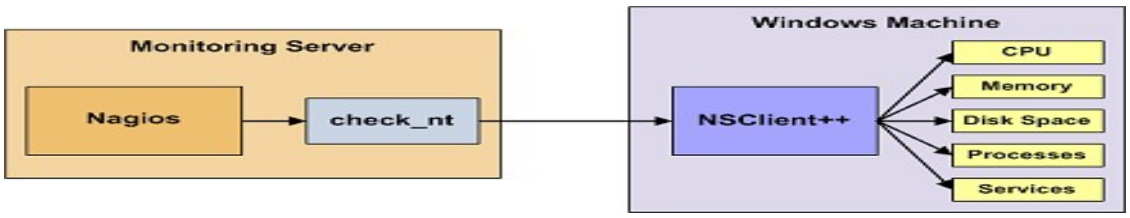
Avec NRPE, la demande d'exécution d'un plugin actif est faite à l'initiative du serveur Nagios. La procédure interne est la suivante:

- le serveur Nagios demande, via le client NRPE, l'exécution du plugin P sur la machine H
- le daemon NRPE hébergé sur la machine H, reçoit la requête d'exécution du plugin P
- le plugin P est exécuté sur la machine H
- le daemon NRPE de la machine H envoie le résultat du plugin P au serveur Nagios
- le serveur Nagios interprète les résultats retournés par le pugin P

Pour l'exécution passive, c'est l'extension NSCA qui est utilisée en lieu et place de NRPE.
NRPE étant déclenché à l'initiative du serveur Nagios, son mode de fonctionnement peut poser problème, par exemple dans le cas ou les machines à surveiller sont derrières un réseau sécurisé par un Firewall. Le plugin NSCA répond à ce problème en proposant l'exécution de plugins passifs sur les machines à surveiller. En effet, la vérification est planifiée en local et le résultat est envoyé au serveur
NSCA est utilisé dans le cadre de la supervision distribuée.
**Note:** Il est possible d'exécuter les plugins nagios sur des machines Linux distantes par SSH (à travers le script *check_by_ssh*). Mais bien que cela soit sécurisé, en contre partie, cela demande plus en charge processeur.

## Supervision des machines Windows

La supervision des machines Windows se fait grâce à l'agent **NSClient++** qui doit être installé sur la machine distante à superviser. Le schéma suivant présente les différents composants qui doivent être mis en place et leur interaction pour que la supervision soit opérationnelle.




Source image: Manuel officiel de nagios

"**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++) doit être installée sur chacune des machines Windows à surveiller.

Le principe de supervision des autres équipements réseaux (Routeurs, Commutateurs, etc...) reste le même.

## La supervision distribuée avec Nagios

Nagios peut être configuré pour supporter la supervision distribuée des services et ressources réseaux. Le but de la supervision distribuée est d'alléger la charge (*CPU, Disque, etc...*) du serveur central de supervision en déléguant certaines tâches de contrôle des services à d'autres serveurs du réseau. Cette technique est intéressante si et seulement si le nombre de machines et services à superviser devient important (*une centaine*).

## Installation de l'extension NRPE

Il s'agit d'une installation classique. Elle se fait sur la machine Linux qui doit être supervisée

```
[root@emabolaptop local]#tar xvzf nrpe-2.12.tar.gz
```

```
[root@emabolaptop local]#cd  nrpe-2.12
[root@emabolaptop nrpe-2.12]#$./configure
```

```
*** Configuration summary for nrpe 2.12 03-10-2008 ***:
 General Options:
 -------------------------
 NRPE port:    5666
 NRPE user:    nagios
 NRPE group:   nagios
 Nagios user:  nagios
 Nagios group: nagios

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

```
[root@emabolaptop nrpe-2.12]# make all
```

```
cd ./src/; make ; cd ..
make[1]: Entering directory `/usr/local/nagios/nrpe-2.12/src'
gcc -g -O2 -I/usr/include/openssl -I/usr/include -DHAVE_CONFIG_H -o nrpe nrpe.c utils.c -L/usr/lib  -lssl -lcrypto -lnsl
gcc -g -O2 -I/usr/include/openssl -I/usr/include -DHAVE_CONFIG_H -o check_nrpe check_nrpe.c utils.c -L/usr/lib  -lssl -lcrypto -lnsl
make[1]: Leaving directory `/usr/local/nagios/nrpe-2.12/src'

*** Compile finished ***

If the NRPE daemon and client compiled without any errors, you
can continue with the installation or upgrade process.

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

```
[root@emabolaptop nrpe-2.12]# make install-plugin
```

```
cd ./src/ && make install-plugin
make[1]: Entering directory `/usr/local/nagios/nrpe-2.12/src'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/libexec
/usr/bin/install -c -m 775 -o nagios -g nagios check_nrpe /usr/local/nagios/libexec
make[1]: Leaving directory `/usr/local/nagios/nrpe-2.12/src'
```

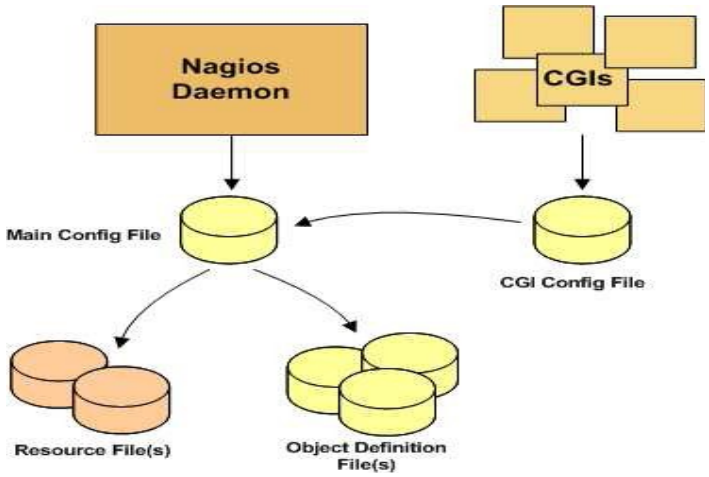
## Configuration de Nagios

Afin de monitorer les machines distantes (services et ressources), nous avons modifié certains fichiers de configuration de nagios. Il s'agit des fichiers suivants:

- Fichier des ressources (***ressource.cfg***): Contenant les macros définies par l'administrateurs
- Fichiers de définition des objets: Ces fichiers sont utilisés pour définir les hôtes, services, groupes d'hôtes, contacts, commandes, etc...
- Fichier de configuration de CGI: Ce fichier contient un certain nombre de directives ayant des effets sur les opérations CGI

L'ensemble de ces fichiers se trouvent dans le répertoire "**`/usr/local/nagios/etc`**" Mais il est possible que ce répertoire ne soit pas le même pour toutes les mises en place de Nagios.

Le schéma ci-dessous présente l'interaction entre le fichier ce configuration principal de Nagios (***nagios.cfg***) et les autres fichiers de configuration.




Source image: Manuel officiel de nagios

Le fichier de configuration principal de Nagios comporte plus d'une centaines d'options. Nous n'allons pas les décrire toutes dans ce document. Nous décrivons juste quelques options, et pour le reste, nous vous renvoyons dans le manuel officiel de Nagios téléchargeable sur le site officiel de Nagios (http://www.nagios.org)

## Fichier de configuration principal de nagios

Dans notre cas, nous avons modifié certaines options dont voici les plus importantes:

| | |
|---|---|
| **log_file** | Permet de préciser où nagios créera son fichier journal principal (/usr/local/nagios/var/nagios.log) |
| **cfg_file** | Directive utilisée pour spécifier le fichier de configuration d'un objet (host, service, etc.) |
| **object_cache_file** | Directive permettant de spécifier un fichier dans lequel seront enregistrées les copies des objets. |
| **resource_file** | Directive permettant de spécifier un fichier optionnel de resources contenant la définition des macros $USERn$ |
| **status_update_interval** | Spécifie comment Nagios doit mettre à jour les données dans les fichiers d'état |
| **nagios_user** | Spécifie l'utilisation sous lequel le démon Nagios s'exécutera (“nagios” dans notre cas) |
| **nagios_group** | Spécifie le groupe sous lequel le démon Nagios s'exécutera (“nagios” dans notre cas) |
| **enable_notifications** | Permet de spécifier si oui ou non, nagios envera les notifications quand il redémarrera initialement |
| **execute_service_checks** | Permet de spécifier si oui ou non, nagios vérifiera les services quand il démarrera ou redémarrera initialement. |

Note: Cette liste d'options n'est pas exhaustive.

Pour qu'un objet soit monitoré, il doit être définit dans un fichier d'objet. Par exemple si c'est un hôte, une entrée dans figurer dans le fichier “**hosts.cfg**” pour cette machine.

Exemple d'entrée:

```
define host{
      host_name                tonton
      use                      generic-host
      alias                    Tonton-Linux
```

```
        address                 192.168.1.20
        _SNMPCOMMUNITY          public
        _SNMPVERSION            3
        parents                 Centreon-Server
        hostgroups              Linux_Servers
        check_command           check_http!$USER2$!80!90
        max_check_attempts      3
        check_interval          60
        active_checks_enabled   1
        passive_checks_enabled        1
        contact_groups                Supervisors
        contacts                      MABO_Elie, Supervisor
        notification_interval         60
        notification_period           24x7
        notification_options          d,u
        notifications_enabled         1
}
```

Pas besoin d'expliquer les différentes variables utilisées dans cette structure, car leur libellé est assez explicatif.

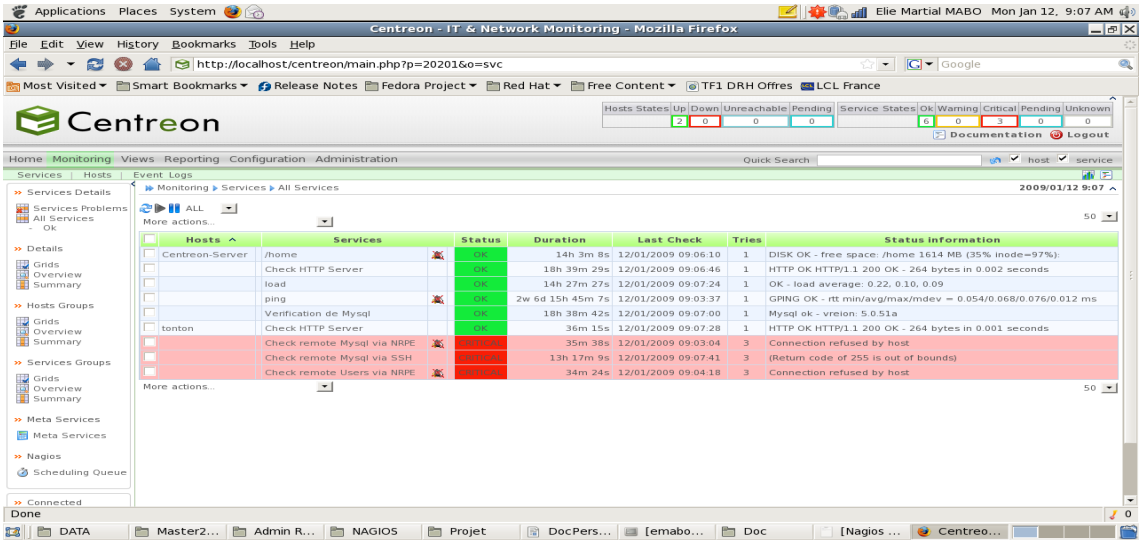
Nous avons donc modifié ou crée un ensemble de fichiers de configurations en fonction de nos besoins, qui étaient juste de monitorer certains services et ressources d'une machine Linux et d'une machine Windows.
L'ensemble des greffons se trouvent dans le répertoire "**/usr/local/nagios/libexec**"

## Développement de ses propres greffons

Les greffons fournis avec Nagios sont destinés à la supervision des services et ressources classiques du style (HTTP, FTP, CPU, etc...). Pour les nouvelles applications, on est obligé de développer ses propres greffons et de les faires exécuter soit par NRPE, soit par NSCA, soit par SSH. Le langage de développement peut être le C, perl, shell, etc....
Dans notre cas, nous avons dévéloppé un greffon qui permet de savoir si la connexion à un base de données mysql a réussi ou pas. Ce greffon a été dévéloppé en langage C. Le code source se trouve en annexe A.

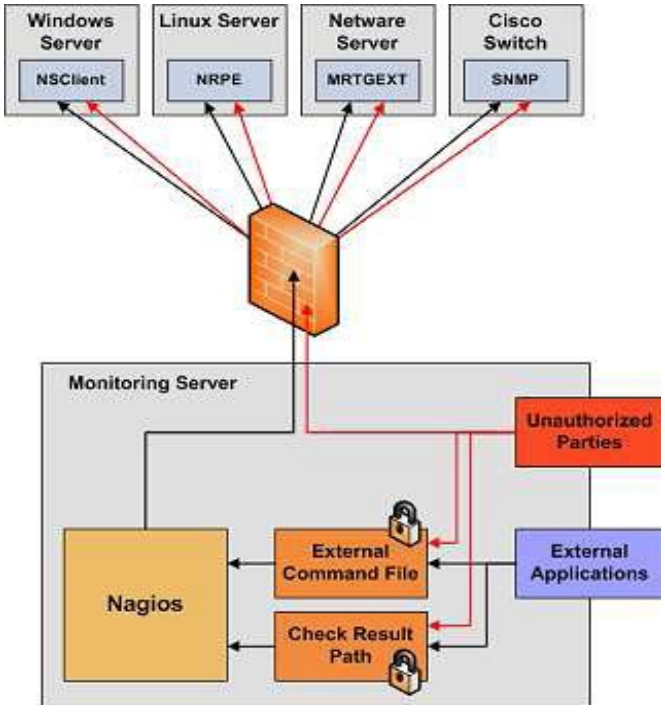
## Combinaison de Nagios et Centreon

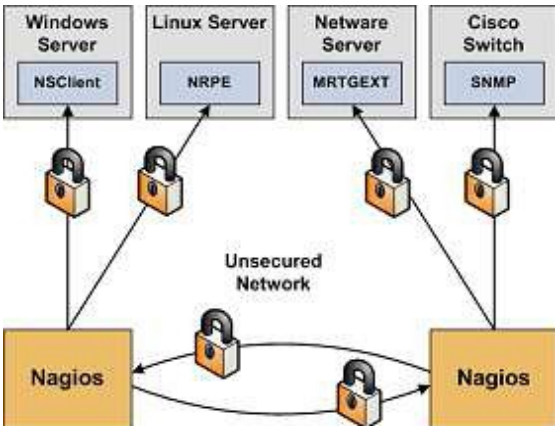
Il est possible d'installer Centreon (*anciennement appelé Oreon*) par dessus Nagios. Centreon est une application open source et gratuit qui permet de gérer Nagios et ses fichiers de configuration via une interface Web. L'avantage de Centreon est qu'elle dispose d'une interface plus intuitive et conviviale.
Dans notre cas, nous avons installé Centreon afin de nous faciliter la gestion de Nagios. Mais nous ne parlerons par de la procédure de mise en place dans ce document.
Voici à quoi ressemble l'interface Web de Centreon:




## Sécurisation de Nagios

Dans un environnement de production, il est fortement déconseillé de déployer un système de supervision si la sécurité de ce dernier ne doit pas être assurée. Cela augmenterait les risques de compromission du système d'information. Par défaut, les échanges entre les serveurs Nagios et les machines à superviser étant en clair sur le réseau, imaginez un event handler que vous avez mis en place pour redémarrer un service en cas d'un état “Warning” par exemple.




**Source images:** Manuel officiel de nagios

Un pirate peut se mettre entre un serveur Nagios et une machine supervisée, et envoyer des codes de retour érronés au serveur. Celui ci passera son temps à redémarrer le service, ce qui est problématique.

Il est donc conseillé pour le déploiement de Nagios, de respecter un ensemble de bonnes pratiques (Best practice) et selon les schémas ci-dessus pour assurer sa sécurisation. Voici quelques unes de ces bonnes pratiques:

- Utilisation des serveurs de supervision dédiés
- Eviter d'exécuter Nagios sous l'identité “root”
- S'assurer que seul l'utilisateur “nagios” est capable de lire et écrire dans le répertoire des résultats
- Exiger l'authentification pour l'accès aux CGI
- Cacher les informations sensives avec les macros $USERSn$
- Sécuriser les accès aux agents (NRPE, NSCA, etc.) distants
- Sécuriser les canaux de communication entre le serveur Nagios et les agents de communication, par exemple en cryptant ces communications.
- Etc.

## Conclusion

Ce travail effectué dans le cadre d'un projet académique nous a permis de comprendre les concepts de la supervision dans un système d'information et de mettre en évidence les différentes architectures possibles dans le cadre de la supervision. Le choix de Nagios et Centreon comme logiciels cobails nous a permis de se mettre en situation réel en s'imaginant dans le monde professionnel. Ces deux logiciels compatibles et matures étant très utilisés dans le monde professionnel. Nous pensons que ce document nous aidera dans un futur proche si jamais nous sommes appelés à travailler dans un projet informatique prenant en compte la supervision.
Il pourrait également aider certains professionnels réseaux et systèmes.

## Webographie et Bibliographie


- http://nagios-fr.org/2008/12/nagios-306
- Nagios: http://www.nagios.org
- Manuel officiel de Nagios (Nagios-3.x)


## Glossaire de termes techniques

| | |
|---|---|
| **CPU** | Central Processing Unit |
| **NRPE:** | Nagios Remote Plugin Executor |
| **NCSA:** | Nagios Service Check Acceptor |
| SSH | Secure SHell |

## Annexe A: Code source en C du greffon permettant d'avoir un état sur la disponibilité d'une base de données mysql.

```
#include <mysql/mysql.h>
#include <stdlib.h>
#include <stdio.h>

#define STATE_CRITCAL 2
#define STATE_WARNING 1
#define STAE_OK 0
MYSQL mysql;

int main (int argc, char ** argv)
{
        uint i=0;
        char *host;
        char *user;
        char *passwd;

        char *status;
        char *version;
        MYSQL mysql;

mysql_init(&mysql);

if(! (mysql_real_connect(&mysql,"localhost","tonton","tonton","base_PNT",0,NULL,0))) {
        printf("Echec connexion à la base Mysql sur la machine : %s \n", host);
        return STATE_CRITCAL;
}

if(!(version= mysql_get_server_info(&mysql))) {
        printf("Connexion réussie, mais impossible d'obtenir des infos du seveur,...,un truc bizarre!\n");
        return STATE_WARNING;
}

printf("Mysql ok - vreion: %s \n", version);
mysql_close(&mysql);
return STAE_OK;
}
```

### Script shell associé

```
#!/bin/bash
STATE_OK=0
STATE_WARNING=1
STATE_CRITICAL=2
/etc/init.d/mysql status >/dev/null
STATE=$?
if test "$STATE" -eq "0" ;
then
echo "TESTD OK"
exit 0
else
echo "TESTD failed"
exit $STATE_CRITICAL
fi
```