

# Installation et configuration de Nagios

**Emmanuel Saracco**

Easter-eggs

[esaracco@easter-eggs.com](mailto:esaracco@easter-eggs.com)

[esaracco@free.fr](mailto:esaracco@free.fr)

## **Installation et configuration de Nagios**

par Emmanuel Saracco

Copyright © 2003-2005 Emmanuel Saracco, Easter-eggs

*Installation et configuration de Nagios*

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is available on the World Wide Web at <http://www.gnu.org/licenses/fdl.html>.

### Historique des versions

Version v1.0a 2005-12-11 Revu par : es

Nommage systématique des chapitres et sections. Mise en place d'une nouvelle feuille de style. Modification du texte de licence.

Version v1.0 2003-11-07 Revu par : es

# Table des matières

<b>1. Introduction.....</b>	<b>1</b>
<b>2. Installation.....</b>	<b>2</b>
<b>3. Paramétrage .....</b>	<b>3</b>
3.1. Apache .....	3
3.2. Gestion des utilisateurs .....	3
3.2.1. Edition du fichier <code>/etc/nagios/cgi.cfg</code> .....	3
3.3. Création des objets .....	4
3.3.1. Gestion des contacts .....	4
3.3.1.1. Création des contacts .....	4
3.3.1.2. Création des groupes de contacts .....	5
3.3.2. Gestion des machines .....	6
3.3.2.1. Création des machines .....	6
3.3.2.1.1. Création d'un modèle (template).....	6
3.3.2.1.2. Création des machines basée sur un modèle .....	7
3.3.2.2. Création des groupes de machines .....	7
3.3.3. Gestion des services à surveiller.....	8
3.3.3.1. Création d'un modèle (template) .....	8
3.3.3.2. Création des services basée sur un modèle .....	9
3.3.4. Vérification de la configuration .....	10
<b>4. Personnalisation .....</b>	<b>12</b>
<b>5. Premiers pas .....</b>	<b>13</b>

# Liste des exemples

3-1. Un contact.....	5
3-2. Un groupe de contacts .....	6
3-3. Un modèle de machine .....	6
3-4. Une machine.....	7
3-5. Un groupe de machines .....	8
3-6. Un modèle de service .....	9
3-7. Un service.....	9

# Chapitre 1. Introduction

Nagios permet de surveiller le bon fonctionnement des services d'une ou plusieurs machines. Il est écrit en C et fonctionne grâce à un ensemble de plugins (qui eux peuvent être écrits dans n'importe quel langage).

Il permet:

- De surveiller les services réseau (SMTP, POP3, HTTP, PING, etc.)
- Une interface permettant l'intégration simple de plugins
- De prévenir par email ou par tout autre méthode personnalisée en cas de problème
- Déclencher des procédures personnalisées pour résoudre les problèmes
- La consultation des différents événements et données collectés via une interface web
- De mettre en place un archivage automatique des données collectées

La distribution GNU/Linux Debian/unstable contient les paquets suivants:

- `nagios-text`
- `nagios-mysql`
- `nagios-pgsql`

Pour une installation sur Debian/Woody, faire une recherche sur [apt-get.org](http://www.apt-get.org)  
(<http://www.apt-get.org/search.php?query=nagios&submit=&arch%5B%5D=i386&arch%5B%5D=all>)

Nagios peut en effet gérer ses données en base de données (MySQL ou PostgreSQL) ou bien dans des fichiers textes.

Il existe un plugin webmin pour nagios (nagmin (<http://nagmin.sourceforge.net/>)) qui fonctionne avec la version packagée pour MySQL.

Nous ne traiterons dans ce document que de la version fichiers textes de nagios. Juste un mot à propos des versions base de données: une fois l'installation du paquet faite, il faudra créer la base de données manuellement en se servant des informations disponibles sous `/usr/share/doc/nagios*`.

# Chapitre 2. Installation

L'installation de la version fichiers textes est simple:

```
# apt-get install nagios-text
```

Par défaut, nagios est installé avec des fichiers de configuration préremplis. Il faudra donc les éditer pour les adapter à notre convenance.

# Chapitre 3. Paramétrage

L'installation de base du paquet ne permet pas d'utiliser de suite nagios. Il est nécessaire de faire un certain nombre de paramétrages pour cela.

## 3.1. Apache

Le paquet installe la configuration apache dans le fichier `/etc/nagios/apache.conf`. Ce fichier est destiné à être inclu dans le fichier de configuration général d'apache. L'inclusion n'est pas faite lors de l'installation. Il faut donc éditer le fichier `/etc/apache/httpd.conf` et y ajouter la ligne suivante:

```
Include /etc/nagios/apache.conf
```

Ensuite il faut relancer apache:

```
/etc/init.d/apache restart
```

## 3.2. Gestion des utilisateurs

Pour que les utilisateurs aient le droit d'accéder à l'interface web il faut tout d'abord créer un fichier d'accès et y ajouter les comptes nécessaires. La configuration par défaut de nagios s'attend à trouver un fichier `/etc/nagios/htpasswd.users`. Nous le créons donc avec un compte d'administration par défaut:

```
# htpasswd -c /etc/nagios/htpasswd.users nagiosadmin
New password: [mot de passe pour nagiosadmin]
Re-type new password: [confirmation du mot de passe]
```

Nous pouvons à présent nous loguer sur l'interface web avec le nouveau compte que nous venons de créer.

Par contre, si nous pouvons nous loguer sans problème, nous n'avons pas les droits sur grand chose... Il va donc falloir cette fois-ci éditer la configuration propre à nagios.

### 3.2.1. Edition du fichier `/etc/nagios/cgi.cfg`

Par défaut, un utilisateur aura le droit d'accéder aux objets auxquels il est associé (nous verrons la création des différents objets plus loin). Si l'on veut que certains utilisateurs aient le droit de visualiser des informations auxquelles ils ne sont pas associés il faudra modifier selon nos besoins les variables de configuration suivantes:

- `authorized_for_system_information`
- `authorized_for_configuration_information`
- `authorized_for_system_commands`
- `authorized_for_all_services`
- `authorized_for_all_hosts`
- `authorized_for_all_service_commands`
- `authorized_for_all_host_commands`

Pour que notre utilisateur `nagiosadmin` puisse accéder aux services de toutes les machines par exemple, il faudra modifier la variable `authorized_for_all_services` comme suit:

```
authorized_for_all_services=nagiosadmin
```

Pour autoriser l'accès à tous les utilisateurs logués sur l'interface web il suffit de mettre un astérisque `*` au lieu de préciser le login.

**Note :** Il n'est pas nécessaire de redémarrer nagios après ce genre de modification.

## 3.3. Création des objets

Nous devons à présent faire connaître la configuration de notre réseau à nagios. Pour cela nous devons mettre en place des groupes, des réseaux, des machines et des services.

### 3.3.1. Gestion des contacts

Les contacts permettront de mettre en place un certain nombre de paramétrages par défaut ainsi que de connaître l'appartenance des objets créés et les personnes à contacter.



### 3.3.1.1. Création des contacts

La gestion des contacts se fait dans le fichier `/etc/nagios/contacts.cfg`.

L'enregistrement d'un contact se fait dans un bloc:

```
define contact{
}

```

On définit ainsi le nom du contact (identifiant que l'on utilisera par la suite pour faire référence à ce contact dans les autres fichiers de configuration), son alias (identifiant plus humainement compréhensible), les périodes de notification par défaut ainsi que les éléments nécessaires pour communiquer avec ce contact. L'exemple qui se trouve dans le fichier est le suivant:

#### Exemple 3-1. Un contact

```
define contact{
    contact_name           nagios
    alias                  Nagios Admin
    service_notification_period 24x7
    host_notification_period  24x7
    service_notification_options w,u,c,r
    host_notification_options  d,u,r
    service_notification_commands notify-by-email,notify-by-epager
    host_notification_commands  host-notify-by-email,host-notify-by-epager
    email                  nagios-admin@localhost.localdomain
    pager                  pagenagios-admin@localhost.localdomain
}

```

Les variables permettant d'indiquer les périodes sont définies dans le fichier `/etc/nagios/timeperiods.cfg`. Il peut être nécessaire de les adapter en fonction du pays ou des horaires propres à la société pour laquelle nagios est à configurer.

Une fois les contacts créés, nous devons créer des groupes auxquels les associer.

### 3.3.1.2. Création des groupes de contacts

La gestion des groupes se fait dans le fichier `/etc/nagios/contactgroups.cfg`.

L'enregistrement d'un groupe se fait dans un bloc:

```
define contactgroup{

```

```
}
```

On définira ici pour chaque groupe créé un nom de groupe (qui permettra d'identifier le groupe par la suite dans les autres fichiers de configuration), un alias (identifiant plus humainement compréhensible), et une liste de contacts à associer au groupe. L'exemple qui se trouve dans le fichier est le suivant:

### Exemple 3-2. Un groupe de contacts

```
define contactgroup{
    contactgroup_name    router-admins
    alias                 Router Technicians
    members               nagios
}
```

L'ajout de plusieurs utilisateurs dans un groupe se fait simplement en séparant les différents identifiants par une virgule.

A présent que nous avons créé des utilisateurs et des groupes, nous pouvons commencer à mettre en place notre réseau.

## 3.3.2. Gestion des machines

La configuration que nous ferons dans cette section permettra à nagios de connaître la base de machines qu'il devra surveiller par la suite.

Nagios offre la possibilité de créer des modèles qui nous permettront de mettre en place une base de configuration standard et d'éviter la redondance de l'information.

### 3.3.2.1. Création des machines

La gestion des machines se fait dans le fichier `/etc/nagios/hosts.cfg`.

#### 3.3.2.1.1. Création d'un modèle (template)

Que l'on crée un modèle ou une machine la syntaxe est la même. On utilise juste l'option `name` et `register` pour le modèle, et l'option `use` pour la machine.

Dans le modèle on définira le nom du modèle ainsi qu'un certain nombre de variable. L'exemple qui se trouve dans le fichier est le suivant:

**Exemple 3-3. Un modèle de machine**

```
define host{
    name        generic-host
    notifications_enabled 0
    event_handler_enabled 0
    flap_detection_enabled 0
    process_perf_data 1
    retain_status_information 1
    retain_nonstatus_information 1
    register    0
}
```

Après avoir défini nos modèles, nous pouvons ajouter les machines.

**3.3.2.1.2. Création des machines basée sur un modèle**

L'enregistrement d'une nouvelle machine se fait dans un bloc:

```
define host{
}

```

Nous définirons ici le modèle à utiliser, le nom de la machine (celui dont nous nous servirons pour y faire référence dans les fichiers de configuration), un alias (identifiant plus humainement compréhensible), la commande à utiliser pour vérifier que la machine est vivante (référence à une commande définie dans `/etc/nagios/checkcommands.cfg`), l'adresse IP de la machine, la période de notification ainsi qu'un certain nombre d'autres variables. L'exemple qui se trouve dans le fichier est le suivant:

**Exemple 3-4. Une machine**

```
define host{
    use                generic-host
    host_name          gw
    alias               Default Gateway
    address             10.0.0.1
    check_command       check-host-alive
    max_check_attempts 20
    notification_interval 60
    notification_period 24x7
    notification_options d,u,r
}
```

Une fois que nous avons créé nos machines nous pouvons les associer à des groupes.

### 3.3.2.2. Création des groupes de machines

La gestion des groupes se fait dans le fichier `/etc/nagios/hostgroups.cfg`.

L'enregistrement d'un groupe se fait dans un bloc:

```
define hostgroup{
}

```

On définira ici pour chaque groupe créé un nom de groupe (qui permettra d'identifier le groupe par la suite dans les autres fichiers de configuration), un alias (identifiant plus humainement compréhensible), les groupes de contacts à y associer et une liste de machines à associer au groupe. L'exemple qui se trouve dans le fichier est le suivant:

#### Exemple 3-5. Un groupe de machines

```
define hostgroup{
    hostgroup_name    gateways
    alias             Routers and Gateways
    contact_groups    router-admins
    members           gw
}

```

L'ajout de plusieurs machines dans un groupe se fait simplement en séparant les différents identifiants par une virgule.

A présent que nous avons créé des groupes de contacts et des groupes de machines, nous pouvons commencer à mettre définir quels seront les services à surveiller pour chaque machine.

### 3.3.3. Gestion des services à surveiller

La gestion des services se fait dans le fichier `/etc/nagios/services.cfg`. Comme pour les machines il est possible pour les services de définir des modèles.

#### 3.3.3.1. Création d'un modèle (template)

Que l'on crée un modèle ou un service la syntaxe est la même. On utilise juste l'option `name` et `register` pour le modèle, et l'option `use` pour le service.

Dans le modèle on définira le nom du modèle ainsi qu'un certain nombre de variable. L'exemple qui se trouve dans le fichier est le suivant:

### Exemple 3-6. Un modèle de service

```
define service{
  name      generic-service
  active_checks_enabled 0
  passive_checks_enabled 0
  parallelize_check 1
  obsess_over_service 1
  check_freshness 0
  notifications_enabled 0
  event_handler_enabled 0
  flap_detection_enabled 0
  process_perf_data 1
  retain_status_information 1
  retain_nonstatus_information 1
  register 0
}
```

Après avoir défini nos modèles, nous pouvons ajouter les services pour chaque machine.

### 3.3.3.2. Création des services basée sur un modèle

L'enregistrement d'un nouveau service se fait dans un bloc:

```
>
define service{
}
```

Nous définirons ici le modèle à utiliser, l'identifiant de la machine concernée, le nom du service voulu, le groupes de contacts responsable de cet élément ainsi que certaines autres variables.

### Exemple 3-7. Un service

```
define service{
  use                generic-service
  host_name          gw
  service_description PING
  is_volatile        0
  check_period       24x7
  max_check_attempts 3
  normal_check_interval 5
  retry_check_interval 1
  contact_groups     router-admins
}
```

```

notification_interval      240
notification_period        24x7
notification_options       c,r
check_command              check_ping!100.0,20%!500.0,60%
}

```

### 3.3.4. Vérification de la configuration

Pour permettre à l'administrateur de vérifier la cohérence des informations saisies dans les fichiers de configuration on peut utiliser nagios avec l'option verbeuse (`-v`). On lui passe en second argument le fichier de configuration à partir duquel faire les vérification (`/etc/nagios/nagios.cfg` en ce qui nous concerne).

```

Nagios 1.1
Copyright (c) 1999-2003 Ethan Galstad (nagios@nagios.org)
Last Modified: 06-02-2003
License: GPL

```

```
Reading configuration data...
```

```
Running pre-flight check on configuration data...
```

```

Checking services...
  Checked 1 services.
Checking hosts...
  Checked 1 hosts.
Checking host groups...
  Checked 1 host groups.
Checking contacts...
  Checked 1 contacts.
Checking contact groups...
  Checked 1 contact groups.
Checking service escalations...
  Checked 1 service escalations.
Checking host group escalations...
  Checked 0 host group escalations.
Checking service dependencies...
  Checked 0 service dependencies.
Checking host escalations...
  Checked 0 host escalations.
Checking host dependencies...
  Checked 0 host dependencies.
Checking commands...
  Checked 22 commands.
Checking time periods...
  Checked 4 time periods.
Checking for circular paths between hosts...
Checking for circular service execution dependencies...

```

```
Checking global event handlers...
Checking obsessive compulsive service processor command...
Checking misc settings...
```

```
Total Warnings: 0
Total Errors:   0
```

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

Une fois que la vérification est ok, redémarrez nagios.

# Chapitre 4. Personnalisation

Nagios permet de personnaliser chacune des machines configurées en y associant des données étendues. Cette section ne concerne en rien l'activité principale de surveillance, mais elle permet d'enjoliver le rendu de certaines informations.



# Chapitre 5. Premiers pas