Version: 2020/02/21 06:18

SUP101 - Gestion de la Supervision avec Nagios sous Debian 8

Présentation

Nagios (**N**agios **A**in't **G**onna **I**nsist **O**n **S**ainthood), connu à l'origine en tant que **Netsaint** est un logiciel qui permet de superviser un système d'information complet.

En résumé, Nagios :

- existe depuis 1999,
- a été créé par **Ethan Galstad**,
- est présent sur approximativement 30 000 installations en production,
- bénéficie d'une communauté de 250 000 utilisateurs,
- est un moteur qui vient ordonnancer les tâches de supervision,
- fourni une interface web pour avoir une vue d'ensemble du système d'information,
- utilise des plugins pour superviser chaque service ou ressource disponible sur l'ensemble des ordinateurs ou éléments réseaux,
- a besoin que le serveur apache soit installé et configuré.

Préparation

Importez une nouvelle machine virtuelle Debian 8 vierge dans VirtualBox.

Installation sous Debian 8

Installation des Dépendances

Commencez par installer les dépendances de Nagios en utilisant apt :

```
root@debian8:~# apt-get update
root@debian8:~# apt install wget gcc make binutils cpp libpq-dev libmysqlclient-dev libssl1.0.0 libssl-dev pkg-config libgd2-xpm-dev libgd-tools perl libperl-dev libnet-snmp-perl snmp apache2 apache2-utils libapache2-mod-php5 unzip tar gzip
```

Création de l'utilisateur nagios

Créez maintenant l'utilisateur **nagios** ayant un mot de passe **trainee** :

```
root@debian8:~# useradd -m nagios
root@debian8:~# passwd nagios
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

Ajoutez les utilisateurs nagios et apache au nouveau groupe **nagcmd** :

```
root@debian8:~# groupadd nagcmd
root@debian8:~# usermod -a -G nagcmd nagios
root@debian8:~# usermod -a -G nagcmd www-data
```

Téléchargement et Compilation de Nagios

Téléchargez maintenant Nagios :

```
root@debian8:~# wget https://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.2.0/nagios-4.2.0.tar.gz
```

```
--2018-11-19 10:58:22--
https://downloads.sourceforge.net/project/nagios/nagios-4.x/nagios-4.2.0/nagios-4.2.0.tar.gz
Résolution de downloads.sourceforge.net (downloads.sourceforge.net)… 216.105.38.13
Connexion à downloads.sourceforge.net (downloads.sourceforge.net)|216.105.38.13|:443… connecté.
requête HTTP transmise, en attente de la réponse… 302 Found
Emplacement : https://vorboss.dl.sourceforge.net/project/nagios/nagios-4.x/nagios-4.2.0/nagios-4.2.0.tar.gz
[suivant]
--2018-11-19 10:58:24--
https://vorboss.dl.sourceforge.net/project/nagios/nagios-4.x/nagios-4.2.0/nagios-4.2.0.tar.gz
Résolution de vorboss.dl.sourceforge.net (vorboss.dl.sourceforge.net)… 5.10.152.194
Connexion à vorboss.dl.sourceforge.net (vorboss.dl.sourceforge.net)|5.10.152.194|:443… connecté.
requête HTTP transmise, en attente de la réponse… 200 OK
Taille : 11155104 (11M) [application/x-gzip]
Enregistre : «nagios-4.2.0.tar.gz»

nagios-4.2.0.tar.gz
100%[=====================================================================================>]  10,64M  2,69MB/s
ds 3,9s

2018-11-19 10:58:46 (2,69 MB/s) - «nagios-4.2.0.tar.gz» enregistré [11155104/11155104]
```

Dernièrement téléchargez les plugins de Nagios :

```
root@debian8:~# wget https://nagios-plugins.org/download/nagios-plugins-2.1.2.tar.gz
--2018-04-24 12:43:40--  https://nagios-plugins.org/download/nagios-plugins-2.1.2.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 72.14.186.43
Connecting to nagios-plugins.org (nagios-plugins.org)|72.14.186.43|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2695301 (2.6M) [application/x-gzip]
Saving to: ‘nagios-plugins-2.1.2.tar.gz’

nagios-plugins-2.1. 100%[====================>]   2.57M  1.81MB/s   in 1.4s
```

```
2018-04-24 12:43:42 (1.81 MB/s) - ‘nagios-plugins-2.1.2.tar.gz’ saved [2695301/2695301]
```

Désarchivez Nagios :

```
root@debian8:~# tar xvf nagios-4.2.0.tar.gz
```

Compilez Nagios :

```
root@debian8:~# cd nagios-4.2.0/
root@debian8:~/nagios-4.2.0# ./configure --with-command-group=nagcmd
...
Creating sample config files in sample-config/ ...


*** Configuration summary for nagios 4.2.0 08-01-2016 ***:

 General Options:
 -------------------------
        Nagios executable:  nagios
        Nagios user/group:  nagios,nagios
       Command user/group:  nagios,nagcmd
             Event Broker:  yes
        Install ${prefix}:  /usr/local/nagios
    Install ${includedir}:  /usr/local/nagios/include/nagios
                Lock file:  ${prefix}/var/nagios.lock
   Check result directory:  ${prefix}/var/spool/checkresults
           Init directory:  /etc/init.d
  Apache conf.d directory:  /etc/httpd/conf.d
              Mail program:  /usr/bin/mail
                   Host OS:  linux-gnu
          IOBroker Method:  epoll

 Web Interface Options:
 ------------------------
```

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


Review the options above for accuracy.  If they look okay,
type 'make all' to compile the main program and CGIs.
```

Les options qui peuvent être passées au script configure sont :

Option	**Description**	**Valeur par Défaut**
–prefix=<dir>	Spécifie le répertoire où seront stockés les binaires de Nagios.	/usr/local/nagios
–sysconfdir=<dir>	Spécifie le répertoire où seront stockés les fichiers de configuration de Nagios.	[prefix]/etc
–localstatedir=<dir>	Spécifie le répertoire où seront stockés les fichiers de statut de Nagios.	[prefix]/var
–with-nagios-user=<utilisateur>	Spécifie l'utilisateur à utiliser par de daemon Nagios.	nagios
–with-nagios-group=<groupe>	Spécifie le groupe à utiliser par de daemon Nagios.	nagios
–with-mail=<chemin>	Spécifie le chemin vers le MTA.	S/O
–with-httpd-conf=<chemin>	Spécifie le chemin vers le répertoire de configuration d'Apache.	S/O
–with-init-dir=<chemin>	Spécifie le chemin vers le répertoire contenant les scripts de controle du service Nagios.	/etc/rc.d/init.d

Saisissez maintenant la commande **make all** :

```
root@debian8:~/nagios-4.2.0# make all
...
*** Compile finished ***

If the main program and CGIs compiled without any errors, you
can continue with installing Nagios as follows (type 'make'
without any arguments for a list of all possible options):

  make install
     - This installs the main program, CGIs, and HTML files
```

```
  make install-init
     - This installs the init script in /etc/rc.d/init.d

  make install-commandmode
     - This installs and configures permissions on the
       directory for holding the external command file

  make install-config
     - This installs *SAMPLE* config files in /usr/local/nagios/etc
       You'll have to modify these sample files before you can
       use Nagios.  Read the HTML documentation for more info
       on doing this.  Pay particular attention to the docs on
       object configuration files, as they determine what/how
       things get monitored!

  make install-webconf
     - This installs the Apache config file for the Nagios
       web interface

  make install-exfoliation
     - This installs the Exfoliation theme for the Nagios
       web interface

  make install-classicui
     - This installs the classic theme for the Nagios
       web interface


*** Support Notes *******************************************

If you have questions about configuring or running Nagios,
please make sure that you:

     - Look at the sample config files
```

```
     - Read the documentation on the Nagios Library at:
          https://library.nagios.com

before you post a question to one of the mailing lists.
Also make sure to include pertinent information that could
help others help you.  This might include:

     - What version of Nagios you are using
     - What version of the plugins you are using
     - Relevant snippets from your config files
     - Relevant error messages from the Nagios log file

For more information on obtaining support for Nagios, visit:

       https://support.nagios.com

**************************************************************

Enjoy.
```

La dernière sortie nous invite à saisir les commande suivantes :

```
root@debian8:~/nagios-4.2.0# make install
...
*** Main program, CGIs and HTML files installed ***

You can continue with installing Nagios as follows (type 'make'
without any arguments for a list of all possible options):

  make install-init
     - This installs the init script in /etc/init.d

  make install-commandmode
     - This installs and configures permissions on the
```

```
       directory for holding the external command file

  make install-config
     - This installs sample config files in /usr/local/nagios/etc

make[1]: Leaving directory '/root/nagios-4.2.0'
```

```
root@debian8:~/nagios-4.2.0# make install-init
/usr/bin/install -c -m 755 -d -o root -g root /etc/init.d
/usr/bin/install -c -m 755 -o root -g root daemon-init /etc/init.d/nagios

*** Init script installed ***
```

```
root@debian8:~/nagios-4.2.0# make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cfg /usr/local/nagios/etc/cgi.cfg
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource.cfg /usr/local/nagios/etc/resource.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/templates.cfg
/usr/local/nagios/etc/objects/templates.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/commands.cfg
/usr/local/nagios/etc/objects/commands.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/contacts.cfg
/usr/local/nagios/etc/objects/contacts.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/timeperiods.cfg
/usr/local/nagios/etc/objects/timeperiods.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/localhost.cfg
/usr/local/nagios/etc/objects/localhost.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/windows.cfg
/usr/local/nagios/etc/objects/windows.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/printer.cfg
/usr/local/nagios/etc/objects/printer.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/switch.cfg
```

```
/usr/local/nagios/etc/objects/switch.cfg

*** Config files installed ***

Remember, these are *SAMPLE* config files.  You'll need to read
the documentation for more information on how to actually define
services, hosts, etc. to fit your particular needs.
```

```
root@debian8:~/nagios-4.2.0# make install-commandmode
/usr/bin/install -c -m 775 -o nagios -g nagcmd -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw

*** External command directory configured ***
```

```
root@debian8:~/nagios-4.2.0# make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/httpd/conf.d/nagios.conf
/usr/bin/install: cannot create regular file ‘/etc/httpd/conf.d/nagios.conf’: No such file or directory
Makefile:296: recipe for target 'install-webconf' failed
make: *** [install-webconf] Error 1
```

Configurer Apache

Comme vous pouvez constater, la dernière commande a retourné une erreur. Cette erreur est due au fait que le script essaie d'installer le fichier **nagios.conf** dans le répertoire **/etc/httpd/conf.d/**, or ce répertoire existe sous Red Hat mais pas sous Debian.

Il est donc necessaire de procéder à une installation manuelle :

```
root@debian8:~/nagios-4.2.0# a2enmod cgi
Enabling module cgi.
To activate the new configuration, you need to run:
  service apache2 restart
root@debian8:~/nagios-4.2.0# a2enmod auth_basic
Considering dependency authn_core for auth_basic:
```

```
Module authn_core already enabled
Module auth_basic already enabled
```

Créez ensuite le fichier **/etc/apache2/conf-available/nagios.conf.** :

```
root@debian8:~/nagios-4.2.0#  vi /etc/apache2/conf-available/nagios.conf
root@debian8:~/nagios-4.2.0#  cat /etc/apache2/conf-available/nagios.conf

ScriptAlias /nagios/cgi-bin "/usr/local/nagios/sbin"

<Directory "/usr/local/nagios/sbin">
#  SSLRequireSSL
   Options ExecCGI
   AllowOverride None
   <IfVersion >= 2.3>
      <RequireAny>
         Require host 127.0.0.1
     Require host 10.0.2.0/24
         AuthName "Nagios Access"
         AuthType Basic
         AuthUserFile /usr/local/nagios/etc/htpasswd.users
         Require valid-user
      </RequireAny>
   </IfVersion>
   <IfVersion < 2.3>
#     Order allow,deny
#     Allow from all
      Order deny,allow
      Deny from all
      Allow from 127.0.0.1 10.0.2.0/24

      AuthName "Nagios Access"
      AuthType Basic
      AuthUserFile /usr/local/nagios/etc/htpasswd.users
```

```
      Require valid-user
   </IfVersion>
</Directory>

Alias /nagios "/usr/local/nagios/share"

<Directory "/usr/local/nagios/share">
#  SSLRequireSSL
   Options None
   AllowOverride None
   <IfVersion >= 2.3>
      <RequireAny>
         Require host 127.0.0.1
     Require host 10.0.2.0/24
         AuthName "Nagios Access"
         AuthType Basic
         AuthUserFile /usr/local/nagios/etc/htpasswd.users
         Require valid-user
      </RequireAny>
   </IfVersion>
   <IfVersion < 2.3>
#     Order allow,deny
#     Allow from all
      Order deny,allow
      Deny from all
      Allow from 127.0.0.1 10.0.2.0/24

      AuthName "Nagios Access"
      AuthType Basic
      AuthUserFile /usr/local/nagios/etc/htpasswd.users
      Require valid-user
   </IfVersion>
</Directory>
```

Activez le module nagios :

```
root@debian8:~/nagios-4.2.0# a2enconf nagios
Enabling conf nagios.
To activate the new configuration, you need to run:
  service apache2 reload
```

Créez un compte **nagiosadmin** pour pouvoir se connecter à l'interface HTML :

```
root@debian8:~/nagios-4.2.0# htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password: fenestros
Re-type new password: fenestros
Adding password for user nagiosadmin
```

Re-démarrez le service Apache2 :

```
root@debian8:~/nagios-4.2.0# systemctl restart apache2
root@debian8:~/nagios-4.2.0# systemctl status apache2
● apache2.service - LSB: Apache2 web server
   Loaded: loaded (/etc/init.d/apache2)
  Drop-In: /lib/systemd/system/apache2.service.d
           └─forking.conf
   Active: active (running) since Tue 2018-04-24 13:40:33 BST; 8s ago
  Process: 870 ExecStop=/etc/init.d/apache2 stop (code=exited, status=0/SUCCESS)
  Process: 900 ExecStart=/etc/init.d/apache2 start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/apache2.service
           ├─914 /usr/sbin/apache2 -k start
           ├─917 /usr/sbin/apache2 -k start
           ├─918 /usr/sbin/apache2 -k start
           ├─919 /usr/sbin/apache2 -k start
           ├─920 /usr/sbin/apache2 -k start
           └─921 /usr/sbin/apache2 -k start

Apr 24 13:40:33 debian8 apache2[900]: Starting web server: apache2.
```

```
Apr 24 13:40:33 debian8 systemd[1]: Started LSB: Apache2 web server.
```

Installer les Plugins de Nagios

Désarchivez **nagios-plugins-2.1.2.tar.gz** :

```
root@debian8:~/nagios-4.2.0# cd ..
root@debian8:~# tar xvf nagios-plugins-2.1.2.tar.gz
```

Compilez maintenant les plugins :

```
root@debian8:~# cd nagios-plugins-2.1.2/
root@debian8:~/nagios-plugins-2.1.2# ./configure --with-nagios-user=nagios --with-nagios-group=nagios
...
config.status: creating po/Makefile
            --with-apt-get-command: /usr/bin/apt-get
              --with-ping6-command: /bin/ping6 -n -U -w %d -c %d %s
               --with-ping-command: /bin/ping -n -U -w %d -c %d %s
                       --with-ipv6: yes
                      --with-mysql: /usr/bin/mysql_config
                   --with-openssl: yes
                    --with-gnutls: no
              --enable-extra-opts: yes
                       --with-perl: /usr/bin/perl
             --enable-perl-modules: no
                      --with-cgiurl: /nagios/cgi-bin
              --with-trusted-path: /usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
                  --enable-libtap: no

root@debian8:~/nagios-plugins-2.1.2# make
...
make[2]: Entering directory '/root/nagios-plugins-2.1.2/po'
make[2]: Nothing to be done for 'all'.
```

```
make[2]: Leaving directory '/root/nagios-plugins-2.1.2/po'
make[2]: Entering directory '/root/nagios-plugins-2.1.2'
make[2]: Leaving directory '/root/nagios-plugins-2.1.2'
make[1]: Leaving directory '/root/nagios-plugins-2.1.2'

[root@centos7 nagios-plugins-2.1.2]# make install
...
make[1]: Leaving directory '/root/nagios-plugins-2.1.2/po'
make[1]: Entering directory '/root/nagios-plugins-2.1.2'
make[2]: Entering directory '/root/nagios-plugins-2.1.2'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/root/nagios-plugins-2.1.2'
make[1]: Leaving directory '/root/nagios-plugins-2.1.2'
```

A l'aide du plugin **check_http**, vérifiez que l'URL **http://127.0.0.1/nagios/** fonctionne correctement :

```
root@debian8:~/nagios-plugins-2.1.2# /usr/local/nagios/libexec/check_http -H 127.0.0.1 -u /nagios/ -a
nagiosadmin:fenestros
HTTP OK: HTTP/1.1 200 OK - 1289 bytes in 0.853 second response time |time=0.853026s;;;0.000000 size=1289B;;;0
```

Résumé de l'Installation

Pour résumer notre installation, voici les chemins les plus importants ainsi que leurs contenus :

Chemin	Contenu
/usr/local/nagios/bin/	Les binaires de nagios.
/usr/local/nagios/libexec/	Les plugins de Nagios.
/usr/local/nagios/etc/	Les fichiers de configuration de Nagios.
/usr/local/nagios/var/	Les journaux de Nagios.

Les Fichiers de Configuration par Défaut

Les fichiers de configuration de Nagios se terminent en majorité par l'extension **.cfg** et se trouvent dans le répertoire **/usr/local/nagios/etc** et le répertoire **/usr/local/nagios/etc/objects** :

```
root@debian8:~# cd ~
root@debian8:~# ls /usr/local/nagios/etc
cgi.cfg  htpasswd.users  nagios.cfg  objects  resource.cfg
```

```
root@debian8:~# ls /usr/local/nagios/etc/objects/
commands.cfg  localhost.cfg  switch.cfg     timeperiods.cfg
contacts.cfg  printer.cfg    templates.cfg  windows.cfg
```

Une exception à cette règle est le fichier contenant le mot de passe de notre utilisateur **nagiosadmin**, à savoir le fichier **/usr/local/nagios/etc/htpasswd.users** :

```
root@debian8:~# cat /usr/local/nagios/etc/htpasswd.users
nagiosadmin:$apr1$fBDvoFzA$W8ZN6O8H/IJwF3P0uo5rl.
```

/usr/local/nagios/etc/objects/contacts.cfg

Les alertes générées par Nagios sont envoyées à une adresse email spécifiée dans le fichier **/usr/local/nagios/etc/objects/contacts.cfg** :

```
root@debian8:~# cat /usr/local/nagios/etc/objects/contacts.cfg
##############################################################################
# CONTACTS.CFG - SAMPLE CONTACT/CONTACTGROUP DEFINITIONS
#
#
# NOTES: This config file provides you with some example contact and contact
#        group definitions that you can reference in host and service
#        definitions.
```

```
#
#        You don't need to keep these definitions in a separate file from your
#        other object definitions.  This has been done just to make things
#        easier to understand.
#
################################################################################



################################################################################
################################################################################
#
# CONTACTS
#
################################################################################
################################################################################

# Just one contact defined by default - the Nagios admin (that's you)
# This contact definition inherits a lot of default values from the 'generic-contact'
# template which is defined elsewhere.

define contact{
        contact_name                    nagiosadmin     ; Short name of user
    use                 generic-contact         ; Inherit default values from generic-contact template (defined
above)
        alias                           Nagios Admin        ; Full name of user

        email                           nagios@localhost    ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS ******
        }
...
```

Important : Notez que si votre serveur dispose d'un serveur de messagerie tel postfix, vous pouvez indiquer une adresse email valide à la directive **email**. Dans le cas contraire il faut au moins que sendmail soit installé sur votre serveur pour que root puisse recevoir les

messages localement.

A Faire - Modifiez l'adresse email nagios@localhost en root@localhost.

Les contacts sont regroupés dans des groupes Nagios. Consultez la section **CONTACT GROUPS** du même fichier :

```
...
##############################################################################
##############################################################################
#
# CONTACT GROUPS
#
##############################################################################
##############################################################################

# We only have one contact in this simple configuration file, so there is
# no need to create more than one contact group.

define contactgroup{
        contactgroup_name       admins
        alias                   Nagios Administrators
        members                 nagiosadmin
        }
```

Important : Notez que **nagiosadmin** dont l'adresse email est maintenant root@localhost est membre du groupe **admins**.

/usr/local/nagios/etc/objects/templates.cfg

Ouvrez maintenant le fichier **/usr/local/nagios/etc/objects/templates.cfg**. Dans ce fichier figurent des *enregistrements types* pour chaque hôte sur un réseau (serveur Linux, Windows™, imprimante, commutateur etc). Consultez la section *Linux host definition template* :

```
...
# Linux host definition template - This is NOT a real host, just a template!

define host{
    name                linux-server    ; The name of this host template
    use                 generic-host    ; This template inherits other values from the generic-host template
    check_period            24x7        ; By default, Linux hosts are checked round the clock
    check_interval            5         ; Actively check the host every 5 minutes
    retry_interval            1         ; Schedule host check retries at 1 minute intervals
    max_check_attempts        10        ; Check each Linux host 10 times (max)
        check_command                 check-host-alive ; Default command to check Linux hosts
    notification_period        workhours    ; Linux admins hate to be woken up, so we only notify during the day
                              ; Note that the notification_period variable is being overridden from
                              ; the value that is inherited from the generic-host template!
    notification_interval        120        ; Resend notifications every 2 hours
    notification_options        d,u,r       ; Only send notifications for specific host states
    contact_groups            admins        ; Notifications get sent to the admins by default
    register            0        ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL HOST, JUST A TEMPLATE!
    }
...
```

Les directives les plus importantes de ce fichier sont :

Directive	**Déscription**
check-host-alive	Vérifie par un ping scripté si l'hôte est en ligne
notification_interval	Indique que la vérification aura lieu toutes les 120 minutes
notification_period	Indique que la vérification aura lieu 24 heures sur 24, 7 jours sur 7
notification_options	Une serie de lettres - **d** en panne, **u** inaccessible, **r** rétabli

Directive	Déscription
flap_detection_enabled	Permet de ne pas être informés des modifications banales telles les connexions des hôtes
contact_groups	Le groupe de contacts à informer en cas de problème

Important : Notez que l'option **register** a une valeur de **0** indiquant que la section n'est pas active.

Le fichier **/usr/local/nagios/etc/objects/templates.cfg** contient aussi des *enregistrements types* pour des services à surveiller :

```
...
##############################################################################
##############################################################################
#
# SERVICE TEMPLATES
#
##############################################################################
##############################################################################

# Generic service definition template - This is NOT a real service, just a template!

define service{
        name                            generic-service     ; The 'name' of this service template
        active_checks_enabled           1               ; Active service checks are enabled
        passive_checks_enabled          1                  ; Passive service checks are enabled/accepted
        parallelize_check               1               ; Active service checks should be parallelized (disabling
this can lead to major performance problems)
        obsess_over_service             1               ; We should obsess over this service (if necessary)
        check_freshness                 0               ; Default is to NOT check service 'freshness'
        notifications_enabled           1               ; Service notifications are enabled
        event_handler_enabled           1               ; Service event handler is enabled
        flap_detection_enabled          1               ; Flap detection is enabled
        process_perf_data               1               ; Process performance data
        retain_status_information       1               ; Retain status information across program restarts
```

```
        retain_nonstatus_information    1               ; Retain non-status information across program restarts
        is_volatile                     0               ; The service is not volatile
        check_period                    24x7            ; The service can be checked at any time of the day
        max_check_attempts              3             ; Re-check the service up to 3 times in order to determine
its final (hard) state
        normal_check_interval           10          ; Check the service every 10 minutes under normal conditions
        retry_check_interval            2           ; Re-check the service every two minutes until a hard state
can be determined
        contact_groups                  admins          ; Notifications get sent out to everyone in the 'admins'
group
    notification_options        w,u,c,r            ; Send notifications about warning, unknown, critical, and
recovery events
        notification_interval           60          ; Re-notify about service problems every hour
        notification_period             24x7            ; Notifications can be sent out at any time
         register                        0              ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL SERVICE,
JUST A TEMPLATE!
        }


# Local service definition template - This is NOT a real service, just a template!

define service{
    name                local-service         ; The name of this service template
    use                generic-service        ; Inherit default values from the generic-service definition
        max_check_attempts              4           ; Re-check the service up to 4 times in order to determine
its final (hard) state
        normal_check_interval           5           ; Check the service every 5 minutes under normal conditions
        retry_check_interval            1           ; Re-check the service every minute until a hard state can be
determined
        register                        0               ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL SERVICE,
JUST A TEMPLATE!
    }
```

/usr/local/nagios/etc/nagios.cfg

Dernièrement, consultez le fichier de configuration global de nagios **/usr/local/nagios/etc/nagios.cfg**. Les directives actives de ce fichier sont :

```
root@debian8:~# egrep -v '^(#|$)'  /usr/local/nagios/etc/nagios.cfg > /tmp/nagios.cfg
root@debian8:~# cat /tmp/nagios.cfg
log_file=/usr/local/nagios/var/nagios.log
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_file=/usr/local/nagios/etc/objects/localhost.cfg
object_cache_file=/usr/local/nagios/var/objects.cache
precached_object_file=/usr/local/nagios/var/objects.precache
resource_file=/usr/local/nagios/etc/resource.cfg
status_file=/usr/local/nagios/var/status.dat
status_update_interval=10
nagios_user=nagios
nagios_group=nagios
check_external_commands=1
command_file=/usr/local/nagios/var/rw/nagios.cmd
lock_file=/usr/local/nagios/var/nagios.lock
temp_file=/usr/local/nagios/var/nagios.tmp
temp_path=/tmp
event_broker_options=-1
log_rotation_method=d
log_archive_path=/usr/local/nagios/var/archives
use_syslog=1
log_notifications=1
log_service_retries=1
log_host_retries=1
log_event_handlers=1
log_initial_states=0
```

```
log_current_states=1
log_external_commands=1
log_passive_checks=1
service_inter_check_delay_method=s
max_service_check_spread=30
service_interleave_factor=s
host_inter_check_delay_method=s
max_host_check_spread=30
max_concurrent_checks=0
check_result_reaper_frequency=10
max_check_result_reaper_time=30
check_result_path=/usr/local/nagios/var/spool/checkresults
max_check_result_file_age=3600
cached_host_check_horizon=15
cached_service_check_horizon=15
enable_predictive_host_dependency_checks=1
enable_predictive_service_dependency_checks=1
soft_state_dependencies=0
auto_reschedule_checks=0
auto_rescheduling_interval=30
auto_rescheduling_window=180
service_check_timeout=60
host_check_timeout=30
event_handler_timeout=30
notification_timeout=30
ocsp_timeout=5
perfdata_timeout=5
retain_state_information=1
state_retention_file=/usr/local/nagios/var/retention.dat
retention_update_interval=60
use_retained_program_state=1
use_retained_scheduling_info=1
retained_host_attribute_mask=0
retained_service_attribute_mask=0
```

```
retained_process_host_attribute_mask=0
retained_process_service_attribute_mask=0
retained_contact_host_attribute_mask=0
retained_contact_service_attribute_mask=0
interval_length=60
check_for_updates=1
bare_update_check=0
use_aggressive_host_checking=0
execute_service_checks=1
accept_passive_service_checks=1
execute_host_checks=1
accept_passive_host_checks=1
enable_notifications=1
enable_event_handlers=1
process_performance_data=0
obsess_over_services=0
obsess_over_hosts=0
translate_passive_host_checks=0
passive_host_checks_are_soft=0
check_for_orphaned_services=1
check_for_orphaned_hosts=1
check_service_freshness=1
service_freshness_check_interval=60
service_check_timeout_state=c
check_host_freshness=0
host_freshness_check_interval=60
additional_freshness_latency=15
enable_flap_detection=1
low_service_flap_threshold=5.0
high_service_flap_threshold=20.0
low_host_flap_threshold=5.0
high_host_flap_threshold=20.0
date_format=us
illegal_object_name_chars=`~!$%^&*|'"<>?,()=
```

```
illegal_macro_output_chars=`~$&|'"<>
use_regexp_matching=0
use_true_regexp_matching=0
admin_email=nagios@localhost
admin_pager=pagenagios@localhost
daemon_dumps_core=0
use_large_installation_tweaks=0
enable_environment_macros=0
debug_level=0
debug_verbosity=1
debug_file=/usr/local/nagios/var/nagios.debug
max_debug_file_size=1000000
allow_empty_hostgroup_assignment=0
```

Les directives les plus utilisées sont :

Directive	**Description**	**Valeur par Défaut**
log_file	Spécifie l'emplacement du fichier journal	/usr/local/nagios/var/nagios.log
cfg_file	Spécifie un fichier de configuration à lire pour définir des objets.	S/O
resource_file	Spécifie le fichier où sont stockées les définitions des macros additionnels.	/usr/local/nagios/etc/resource.cfg
cfg_dir	Spécifie un répertoire contenant des fichiers de configuration à lire pour définir des objets.	S/O
temp_file	Spécifie un chemin vers le fichier utilisé pour stocker des données temporaires.	/usr/local/nagios/var/nagios.tmp
lock_file	Spécifie un chemin vers le fichier utilisé pour la synchronisation.	/usr/local/nagios/var/nagios.lock
temp_path	Spécifie un répertoire où Nagios peut créer des fichiers temporaires.	/tmp
status_file	Spécifie un chemin vers un fichier contenant les statuts actuels de tous les hôtes et services.	/usr/local/nagios/var/status.dat
status_update_interval	Spécifie l'interval de mise à jour du fichier status_file.	10 secondes
nagios_user	Spécifie l'utilisateur qui exécute le daemon Nagios.	S/O
nagios_group	Spécifie le groupe qui exécute le daemon Nagios.	S/O
command_file	Spécifie le chemin vers la ligne de commande externe utilisée par d'autres processus afin de contrôler le daemon Nagios.	/usr/local/nagios/var/rw/nagios.cmd

Directive	Description	Valeur par Défaut
use_syslog	Spécifie si Nagios doit journaliser vers Syslog en même temps que vers son propre fichier de journalisation.	1 (activé)
state_retention_file	Spécifie le fichier utilisé pour stocker les informatins de statut entre des redémarrages.	/usr/local/nagios/var/retention.dat
retention_update_interval	Spécifie l'interval de mise à jour du fichier state_retention_file.	60 secondes
service_check_timeout	Spécifie la période après laquelle une vérification de service retourne le résultat “echec”.	60 secondes
host_check_timeout	Spécifie la période après laquelle une vérification de hôte retourne le résultat “echec”.	60 secondes
event_handler_timeout	Spécifie la durée de vie d'un 'event handler'.	30 secondes
notification_timeout	Spécifie la période après laquelle une tentative de notification retourne le résultat “echec”.	30 secondes
enable_environment_macros	Spécifie si Nagios doit passer les macros aux plugins en tant que variables d'environnement.	1 (activé)
interval_length	Spécifie la valeur d'un “unit interval”.	60 secondes

Important : Notez que la directive **log_file doit être** sur la première ligne du fichier.

Important - Les définitions des directives du fichier **/usr/local/nagios/etc/nagios.cfg** peuvent être trouvées à cette **adresse Internet**

Afin de vérifier la configuration de Nagios avant le démarrage du service, il convient d'utiliser la commande suivante :

```
root@debian8:~# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.2.0
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 08-01-2016
```

```
License: GPL

Website: https://www.nagios.org
Reading configuration data...
   Read main config file okay...
   Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
    Checked 8 services.
    Checked 1 hosts.
    Checked 1 host groups.
    Checked 0 service groups.
    Checked 1 contacts.
    Checked 1 contact groups.
    Checked 24 commands.
    Checked 5 time periods.
    Checked 0 host escalations.
    Checked 0 service escalations.
Checking for circular paths...
    Checked 1 hosts
    Checked 0 service dependencies
    Checked 0 host dependencies
    Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors:   0

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

/usr/local/nagios/etc/resource.cfg

Le fichier **/usr/local/nagios/etc/resource.cfg** contient les définitions des macros **$USERx$**. Ces macros, d'un nombre total de 32, ne peuvent pas être consultées à partir de l'interface web de Nagios et, par conséquence, peuvent être utilisées pour stocker des informations sensibles telles les mots de passe :

```
[root@nagios ~]# cat /usr/local/nagios/etc/resource.cfg
##############################################################################
#
# RESOURCE.CFG - Sample Resource File for Nagios 4.2.0
#
#
# You can define $USERx$ macros in this file, which can in turn be used
# in command definitions in your host config file(s).  $USERx$ macros are
# useful for storing sensitive information such as usernames, passwords,
# etc.  They are also handy for specifying the path to plugins and
# event handlers - if you decide to move the plugins or event handlers to
# a different directory in the future, you can just update one or two
# $USERx$ macros, instead of modifying a lot of command definitions.
#
# The CGIs will not attempt to read the contents of resource files, so
# you can set restrictive permissions (600 or 660) on them.
#
# Nagios supports up to 32 $USERx$ macros ($USER1$ through $USER32$)
#
# Resource files may also be used to store configuration directives for
# external data sources like MySQL...
#
##############################################################################

# Sets $USER1$ to be the path to the plugins
$USER1$=/usr/local/nagios/libexec
```

```
# Sets $USER2$ to be the path to event handlers
#$USER2$=/usr/local/nagios/libexec/eventhandlers

# Store some usernames and passwords (hidden from the CGIs)
#$USER3$=someuser
#$USER4$=somepassword
```

Démarrage de Nagios

Activez et démarrez le service nagios :

```
root@debian8:~# service nagios status
● nagios.service
   Loaded: not-found (Reason: No such file or directory)
   Active: inactive (dead)
root@debian8:~# systemctl enable nagios
Synchronizing state for nagios.service with sysvinit using update-rc.d...
Executing /usr/sbin/update-rc.d nagios defaults
Executing /usr/sbin/update-rc.d nagios enable
root@debian8:~# systemctl start nagios
root@debian8:~# systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
   Loaded: loaded (/etc/init.d/nagios)
   Active: active (running) since Tue 2018-04-24 15:06:39 BST; 6s ago
  Process: 5718 ExecStart=/etc/init.d/nagios start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/nagios.service
           ├─5751 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─5753 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─5754 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─5755 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─5756 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─5757 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
```

```
Apr 24 15:06:39 debian8 nagios[5751]: nerd: Channel hostchecks registered successfully
Apr 24 15:06:39 debian8 nagios[5751]: nerd: Channel servicechecks registered successfully
Apr 24 15:06:39 debian8 nagios[5751]: nerd: Channel opathchecks registered successfully
Apr 24 15:06:39 debian8 nagios[5751]: nerd: Fully initialized and ready to rock!
Apr 24 15:06:39 debian8 nagios[5751]: wproc: Successfully registered manager as @wproc with query handler
Apr 24 15:06:39 debian8 nagios[5751]: wproc: Registry request: name=Core Worker 5754;pid=5754
Apr 24 15:06:39 debian8 nagios[5751]: wproc: Registry request: name=Core Worker 5756;pid=5756
Apr 24 15:06:39 debian8 nagios[5751]: wproc: Registry request: name=Core Worker 5753;pid=5753
Apr 24 15:06:39 debian8 nagios[5751]: wproc: Registry request: name=Core Worker 5755;pid=5755
Apr 24 15:06:40 debian8 nagios[5751]: Successfully launched command file worker with pid 5757
```

Vous pouvez obtenir une vue globale de la configuration de Nagios grâce à la commande :

```
root@debian8:~# /usr/local/nagios/bin/nagios -s /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.2.0
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 08-01-2016
License: GPL

Website: https://www.nagios.org
Timing information on object configuration processing is listed
below.  You can use this information to see if precaching your
object configuration would be useful.

Object Config Source: Config files (uncached)

OBJECT CONFIG PROCESSING TIMES      (* = Potential for precache savings with -u option)
----------------------------------
Read:                 0.039211 sec
Resolve:              0.000038 sec  *
Recomb Contactgroups: 0.006227 sec  *
Recomb Hostgroups:    0.000018 sec  *
```

```
Dup Services:         0.000031 sec  *
Recomb Servicegroups: 0.000001 sec  *
Duplicate:            0.000001 sec  *
Inherit:              0.000002 sec  *
Register:             0.000125 sec
Free:                 0.000022 sec
                      ============
TOTAL:                0.045677 sec  * = 0.001580 sec (3.46%) estimated savings


Timing information on configuration verification is listed below.

CONFIG VERIFICATION TIMES
---------------------------------
Object Relationships: 0.000042 sec
Circular Paths:       0.000002 sec
Misc:                 0.000229 sec
                      ============
TOTAL:                0.000273 sec


RETENTION DATA TIMES
---------------------------------
Read and Process:     0.000838 sec
                      ============
TOTAL:                0.000838 sec


EVENT SCHEDULING TIMES
-------------------------------------
Get service info:        0.000087 sec
Get host info info:      0.000006 sec
Get service params:      0.000007 sec
Schedule service times:  0.000146 sec
```

```
Schedule service events: 0.000040 sec
Get host params:         0.000000 sec
Schedule host times:     0.000022 sec
Schedule host events:    0.000005 sec
                         ============
TOTAL:                   0.000313 sec


Projected scheduling information for host and service checks
is listed below.  This information assumes that you are going
to start running Nagios with your current config files.

HOST SCHEDULING INFORMATION
---------------------------
Total hosts:                     1
Total scheduled hosts:           1
Host inter-check delay method:   SMART
Average host check interval:     300.00 sec
Host inter-check delay:          300.00 sec
Max host check spread:           30 min
First scheduled check:           Tue Apr 24 23:20:23 2018
Last scheduled check:            Tue Apr 24 23:20:23 2018


SERVICE SCHEDULING INFORMATION
------------------------------
Total services:                    8
Total scheduled services:          8
Service inter-check delay method:  SMART
Average service check interval:    300.00 sec
Inter-check delay:                 37.50 sec
Interleave factor method:          SMART
Average services per host:         8.00
Service interleave factor:         8
```

```
Max service check spread:           30 min
First scheduled check:              Tue Apr 24 23:21:00 2018
Last scheduled check:               Tue Apr 24 23:25:23 2018


CHECK PROCESSING INFORMATION
----------------------------
Average check execution time:    0.51s
Estimated concurrent checks:     1 (1.00 per cpu core)
Max concurrent service checks:   Unlimited


PERFORMANCE SUGGESTIONS
-----------------------
I have no suggestions - things look okay.
```

Dernièrement, une trace complète de l'activité de nagios peut être trouvé dans **/usr/local/nagios/var/nagios.log** :

```
root@debian8:~# cat /usr/local/nagios/var/nagios.log
[1524578799] Nagios 4.2.0 starting... (PID=5751)
[1524578799] Local time is Tue Apr 24 15:06:39 BST 2018
[1524578799] LOG VERSION: 2.0
[1524578799] qh: Socket '/usr/local/nagios/var/rw/nagios.qh' successfully initialized
[1524578799] qh: core query handler registered
[1524578799] nerd: Channel hostchecks registered successfully
[1524578799] nerd: Channel servicechecks registered successfully
[1524578799] nerd: Channel opathchecks registered successfully
[1524578799] nerd: Fully initialized and ready to rock!
[1524578799] wproc: Successfully registered manager as @wproc with query handler
[1524578799] wproc: Registry request: name=Core Worker 5754;pid=5754
[1524578799] wproc: Registry request: name=Core Worker 5756;pid=5756
[1524578799] wproc: Registry request: name=Core Worker 5753;pid=5753
[1524578799] wproc: Registry request: name=Core Worker 5755;pid=5755
[1524578800] Successfully launched command file worker with pid 5757
```

```
[1524582398] Auto-save of retention data completed successfully.
[1524585998] Auto-save of retention data completed successfully.
[1524589598] Auto-save of retention data completed successfully.
[1524593198] Auto-save of retention data completed successfully.
[1524596798] Auto-save of retention data completed successfully.
[1524600398] Auto-save of retention data completed successfully.
[1524603998] Auto-save of retention data completed successfully.
[1524607598] Auto-save of retention data completed successfully.
```

Interface Web

Testez ensuite que l'interface web de Nagios fonctionne :

```
root@debian8:~# lynx -auth=nagiosadmin:fenestros --dump http://localhost/nagios
   FRAME: [1]side
   FRAME: [2]main

Nagios Core

                              [3]www.nagios.org
      Copyright © 2010-2016 Nagios Core Development Team and Community
             Contributors. Copyright © 1999-2010 Ethan Galstad

   Note: These pages require a browser which supports frames

References

   1. http://localhost/nagios/side.php
   2. http://localhost/nagios/main.php
   3. https://www.nagios.org/
```

Configurer le Réseau

Configurez maintenant une adresse IP fixe pour le serveur Nagios :

```
root@debian8:~# vi /etc/NetworkManager/NetworkManager.conf
root@debian8:~# cat /etc/NetworkManager/NetworkManager.conf
[main]
plugins=ifupdown,keyfile

[ifupdown]
managed=true

root@debian8:~# systemctl restart NetworkManager
root@debian8:~# nmcli c show
NAME             UUID                                  TYPE            DEVICE
eth0             7b83daa8-d1d0-4595-8232-dcdbc4221e92  802-3-ethernet  eth0
Ifupdown (eth0)  681b428f-beaf-8932-dce4-687ed5bae28e  802-3-ethernet  --

root@debian8:~# nmcli connection add con-name ip_fixe ifname eth0 type ethernet ip4 10.0.2.14/24 gw4 10.0.2.2
Connection 'ip_fixe' (36f5c5ba-ccc6-41f3-a286-8be779802aa5) successfully added.
root@debian8:~# nmcli connection mod ip_fixe ipv4.dns 8.8.8.8
root@debian8:~# nmcli g hostname nagios.i2tch.loc
root@debian8:~# nmcli connection up ip_fixe
```

A Faire - Fermez votre session SSH et ouvrez en une autre.

```
root@nagios:~# nmcli c show
NAME             UUID                                  TYPE            DEVICE
ip_fixe          36f5c5ba-ccc6-41f3-a286-8be779802aa5  802-3-ethernet  eth0
eth0             7b83daa8-d1d0-4595-8232-dcdbc4221e92  802-3-ethernet  --
```

```
Ifupdown (eth0)  681b428f-beaf-8932-dce4-687ed5bae28e  802-3-ethernet  --

root@nagios:~# dig www.free.fr

; <<>> DiG 9.9.5-9+deb8u15-Debian <<>> www.free.fr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52590
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.free.fr.           IN    A

;; ANSWER SECTION:
www.free.fr.        12033    IN    A    212.27.48.10

;; Query time: 58 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Sat May 12 13:42:28 BST 2018
;; MSG SIZE  rcvd: 56
```

A Faire : Arrêter votre VM. Augmentez la mémoire allouée à la machine virtuelle à 2 Go. Démarrez votre VM.

Configuration

Afin de faciliter l'utilisation de Nagios, l'organisation des données sera basée sur une arborescence de répertoires où chaque répertoire sera spécifique à un type d'objet et des objets similaires seront regroupés dans le même fichier. Par exemple, les définitions des commandes seront stockées dans le

sous-répertoire **commands/** et les définitions des hôtes dans **hosts/<nom_hôte>.cfg**. Pour mettre en place cette organisation, commencez par éditer le fichier **/tmp/nagios.cfg** en remplacant toutes des lignes commencant par **cfg_file** ou **cfg_dir** avec les lignes suivantes :

```
cfg_dir=/usr/local/nagios/etc/commands
cfg_dir=/usr/local/nagios/etc/timeperiods
cfg_dir=/usr/local/nagios/etc/contacts
cfg_dir=/usr/local/nagios/etc/contactgroups
cfg_dir=/usr/local/nagios/etc/hosts
cfg_dir=/usr/local/nagios/etc/hostgroups
cfg_dir=/usr/local/nagios/etc/services
cfg_dir=/usr/local/nagios/etc/servicegroups
```

par exemple :

```
root@nagios:~# vi /tmp/nagios.cfg
root@nagios:~# cat /tmp/nagios.cfg
log_file=/usr/local/nagios/var/nagios.log
#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_file=/usr/local/nagios/etc/objects/localhost.cfg
cfg_dir=/usr/local/nagios/etc/commands
cfg_dir=/usr/local/nagios/etc/timeperiods
cfg_dir=/usr/local/nagios/etc/contacts
cfg_dir=/usr/local/nagios/etc/contactgroups
cfg_dir=/usr/local/nagios/etc/hosts
cfg_dir=/usr/local/nagios/etc/hostgroups
cfg_dir=/usr/local/nagios/etc/services
cfg_dir=/usr/local/nagios/etc/servicegroups
object_cache_file=/usr/local/nagios/var/objects.cache
...
allow_empty_hostgroup_assignment=0
```

Créez maintenant les répertoires referencés par le fichier de configuration :

```
root@nagios:~# cd /usr/local/nagios/etc/
root@nagios:/usr/local/nagios/etc# mkdir commands timeperiods contacts contactgroups hosts hostgroups services
servicegroups
root@nagios:/usr/local/nagios/etc# ls
cgi.cfg        contacts    htpasswd.users  resource.cfg   timeperiods
commands       hostgroups  nagios.cfg      servicegroups
contactgroups  hosts       objects       services
root@nagios:/usr/local/nagios/etc# cd ~
```

Afin d'utiliser les plugins de Nagios, copiez le fichier des définitions de commandes **/usr/local/nagios/etc/objects/commands.cfg** vers **/usr/local/nagios/etc/commands/default.cfg** :

```
root@nagios:~# cp /usr/local/nagios/etc/objects/commands.cfg /usr/local/nagios/etc/commands/default.cfg
```

Utilisation de Macros

Les Macros Standards

Toutes les commandes de Nagios peuvent utiliser des macros. Les définitions des macros permettent le réferencment de paramètres d'autres objets, tels les hôtes, les services et les contacts de façon à éviter de tout passer à la commande en tant qu'argument. Chaque invocation d'un macro doit commencer avec le caractère **$** et se terminer avec le caractère **$** :

```
  define host{
    host_name        somemachine
    address          10.0.0.1
    check_command    check-host-alive
    }

  define command{
    command_name     check-host-ssh
```

```
    command_line     $USER1$/check_ssh -H $HOSTADDRESS$
    }
```

Dans le cas du macro ci-dessus, la commande suivante serait invoquée :

```
/usr/local/nagios/libexec/check_ssh -H 10.0.0.1
```

Les macros les plus utilisés sont :

Macro	**Description**
HOSTNAME	Le nom court et unique de l'hôte. Correspond à la directive **host_name** dans l'objet hôte.
HOSTADDRESS	L'adresse IP ou le nom d'hôte de l'hôte. Correspond à la directive **address** dans l'objet hôte.
HOSTDISPLAYNAME	La description de l'hôte. Correspond à la directive **alias** dans l'objet hôte.
HOSTSTATE	L'état actuel de l'hôte parmi UP, DOWN ou UNREACHABLE.
HOSTGROUPNAMES	Les noms courts, séparés par des virgules, de tous les groupes dont l'hôte est membre.
LASTHOSTCHECK	La date et l'heure de la dernière vérification de l'hôte exprimées au format horodatage UNIX, à savoir le nombre de secondes depuis 1970-01-01.
LASTHOSTSTATE	Le dernier état connu de l'hôte parmi UP, DOWN ou UNREACHABLE.
SERVICEDESC	La description du service. Correspond à la directive **description** dans l'objet service.
SERVICESTATE	L'état actuel du service parmi OK, WARNING, UNKOWN ou CRITICAL.
SERVICEGROUPNAMES	Les noms courts, séparés par des virgules, de tous les groupes de services dont le service est membre.
CONTACTNAME	Le nom court et unique du contact. Correspond à la directive **contact_name** dans l'objet contact.
CONTACTALIAS	La description du contact. Correspond à la directive **alias** dans l'objet contact.
CONTACTEMAIL	L'adresse email du contact. Correspond à la directive **email** dans l'objet contact.
CONTACTGROUPNAMES	Les noms courts, séparés par des virgules, de tous les groupes de contacts dont le contact est membre.

Les Macros sur Demande

Certains macros permettent l'utilisation d'un ou de plusieurs arguments encapsulés afin de référencer un autre objet. Par exemple **$CONTACTEMAIL:jdoe$** permet d'obtenir l'adresse email de l'utilisateur jdoe.

Les Macros d'Utilisateur

Il est aussi possible pour l'administreur de définir des attributs additionnels dans chaque type de macro. Les informations stockées peuvent ensuite être référencées ainsi :

- $_HOST<variable>$ - pour les directives définies dans un objet hôte,
- $_SERVICE<variable>$ - pour les directives définies dans un objet service,
- $_CONTACT<variable>$ - pour les directives définies dans un objet contact.

L'attribut doit commencer avec le caractère _ et être en majuscules, tel **_MAC** dans l'exemple suivant :

```
define host{
  host_name        somemachine
  address          10.0.0.1
  _MAC             12:12:12:12:12:12
  check_command    check-host-by-mac
  }
```

Une commande de vérification y fera référence de la façon suivante :

```
define command{
  command_name     check-host-by-mac
  command_line     $USER1$/check_hostmac -H $HOSTADDRESS$ -m
    $_HOSTMAC$
  }
```

Il est suggéré de préfixer le nom de l'attribut avec deux caractères _ afin d'améliorer la lisibilité par la suite :

```
define host{
  host_name        somemachine
  address          10.0.0.1
  __MAC            12:12:12:12:12:12
  check_command    check-host-by-mac
```

```
}
```

Dans ce cas, la commande y fera référence ainsi :

- $_HOST_MAC$

Variables d'Environnement

La majorité des attributs de macros standards sont exportés en tant que variables d'environnement si la directive **enable_environment_macros** du fichier **/usr/local/nagios/etc/nagios.cfg** n'est **pas** configuré ainsi :

- enable_environment_macros=0

Important : Il est à noter que les atttributs des macros sur demande et des macros utilisateurs ne sont **pas** exportés en tant que variables d'environnement.

Il est ensuite possible de faire référence à l'attribut en faisant précéder le nom de celui-ci par **NAGIOS_**. Par exemple dans le cas de l'attribut **HOSTADDRESS** :

- **NAGIOS_HOSTADDRESS**

La Configuration des Hôtes

Un hôte est un objet qui décrit une machine, physique ou virtuelle, qui consiste en :

- un nome court unique,
- un nom desciptif,
- une adresse IP ou nom d'hôte,
- quand et comment le système doit être surveillé,
- à quelle fréquence le système doit être surveillé,

- comment gérer les vérifications multiples,
- qui doit être contacté en cas de problème,
- comment seront envoyés les notifications concernant les problèmes.

Par exemple :

```
define host{
  host_name                       linuxbox01
  hostgroups                      linuxservers
  alias                           Linux Server 01
  address                         10.0.2.15
  check_command                   check-host-alive
  check_interval                  10
  retry_interval                  1
  max_check_attempts              5
  check_period                    24x7
  contact_groups                  linux-admins
  notification_interval           30
  notification_period             24x7
  notification_options            d,u,r
  }
```

Dans le cas ci-dessus le test aura lieu tous les 10 minutes. Après 5 tests en échec, le système sera marqué comme “couché”. Quand le système est couché, une notification sera envoyée toutes les 30 minutes.

Le tableau suivant résume les directives usuelles pour décrire un hôte :

Directive	**Description**
host_name	Le nom court et unique de l'hôte.
alias	Un nom descriptif de l'hôte.
address	L'adresse IP ou le FQDN de l'hôte. Il est préférable d'utiliser une adresse IP afin d'éviter que les tests soient en échec lors d'une panne de serveur DNS.
parents	Une liste des hôtes parents, séparés par des virgules, dont l'hôtre courant est dépendant. En règle générale, ceci est un switch ou un routeur.

Directive	Description
hostgroups	Une liste de tous les groupes, séparés par des virgules, dont l'hôte courant devait être membre.
check_command	Le nom court de la commande qui devrait être utilisée pour vérifier si le hôte courant fonctionne. Si la commande retourne un état OK, l'état de l'hôte est interpreter comme UP sinon l'état est considéré d'être DOWN.
check_interval	Spécifie en minutes la fréquence des tests.
retry_interval	Spécifie le nombre de minutes à attendre avant de procéder de nouveau à un test dans le cas où l'état est DOWN.
max_check_attempts	Spécifie le nombre de tests à éffectuer avant que Nagios considère que l'hôte est DOWN.
check_period	Spécifie le nom de la plage temporelle utilisée pour déterminer les plages d'horaires pendant lesquelles les tests sont effectués.
contacts	Une liste de contacts, séparés par des virgules, qui doivent être contactés lors d'un changement de l'état de l'hôte.
contact_groups	Une liste de groupe de contacts, séparés par des virgules, qui doivent être contactés lors d'un changement de l'état de l'hôte.
first_notification_delay	Spécifie le nombre de minutes à attendre avant que la première notification d'un changement d'état soit envoyée.
notification_interval	Spécifie la fréquence de l'envoi des notifications.
notification_period	Spécifie les plages temporelles penadnt lesquelles les notifications seront envoyées.
notification_options	Une liste de type de notification d'états, séparés par des virgules, qui déclencheront une notification. Doit être un ou plusieurs des lettres suivants : **d** - DOWN, **u** - UNREACHABLE, **r** - recovery (UP), **f** - FLAPPING (oscillation d'état) ou **s** - scheduled downtime (début ou fin d'un arrêt planifié.

Important : Au moins un contact ou un groupe de contacts doit être spécifié par hôte.

Par défaut Nagios présume que l'état de tous les hôtes est **UP**. De cette façon si l'option **check_command** n'est pas spécifiée, l'hôte sera toujours considéré comme étant dans un état **UP**.

Nagios utilise une logique d'état de type **soft** et **hard**. Par conséquent si l'état d'un hôte a changé de **UP** à **DOWN** depuis le derneir **état en dur**, Nagios présume que l'état actuel de l'hôte est **soft DOWN**. Il va ensuite ré-essayer le **check_command** chaque **retry_interval** jusqu'à atteindre le **max-check-attempts**. A ce stade Nagios présume que l'état de l'hôte est devenu **hard DOWN**. Les notifications ne sont envoyées que dans le cas où l'état est de type HARD.

Bien évidement le même processus aura lieu dans une transition de DOWN à UP.

La Directive parents de l'objet Hôte

La directive parents de l'objet hôte est utilisée pour définir la topologie du réseau. En règle générale la directive désigne un routeur ou un switch auquel est connecté l'hôte. Si l'état du parent est hard DOWN, les hôtes enfants seront considérés comme étant DOWN et aucun test sur ces objets ne sera effectué.

La Configuration des Groupes d'Hôtes

Nagios permet l'utilisation de groupements d'hôtes appelés des **Groupes** :

- un groupe a un nom court, un nom desciptif ainsi qu'un ou plueirus membres,
- un hôte peut être membre d'un ou de plusieurs groupes,
- en règle générale les groupes regroupent des machines du même type, de la me location géographique ou du même rôle.

Par exemple :

```
define hostgroup{
  hostgroup_name                linux-servers
  alias                         Linux servers
  members                       linuxbox01,linuxbox02
  }

define hostgroup{
  hostgroup_name                aix-servers
  alias                         AIX servers
  members                       aixbox1,aixbox2
  }

define hostgroup{
  hostgroup_name                unix-servers
  alias                         UNIX servers servers
  hostgroup_members             linux-servers,aix-servers
```

```
    }
```

Le tableau suivant résume les directives usuelles pour décrire un groupe d'hôtes :

Directive	**Description**
hostgroup_name	Le nom court et unique du groupe d'hôtes.
alias	Un nom descriptif du groupe d'hôtes.
members	Une liste d'hôtes, séparés par des virgules, qui sont membre du groupe de hôtes courant.
hostgroup_members	Une liste de groupe d'hôtes, séparés par des virgules, dont les membres doivent aussi être membres du groupe de hôtes courant.

La Surveillance de l'Hôte Nagios

Afin de surveiller notre propre hôte Nagios, créez maintenant un hôte dans le répertoire **/usr/local/nagios/etc/hosts** appelé **localhost.cfg** :

```
root@nagios:~# vi /usr/local/nagios/etc/hosts/localhost.cfg
root@nagios:~# cat /usr/local/nagios/etc/hosts/localhost.cfg
  define host{
    host_name                       localhost
    alias                           Localhost
    address                         127.0.0.1
    check_command                   check-host-alive
    check_interval                  5
    retry_interval                  1
    max_check_attempts              5
    check_period                    24x7
    contact_groups                  admins
    notification_interval           60
    notification_period             24x7
    notification_options            d,u,r
    }
```

La Configuration des Services

Un service est un objet qui décrit une fonctionalité que fournit un hôte spécifique.

Un service :

- est toujours lié à un hôte spécifique,
- est identifié par une déscription unique dans le même hôte,
- définit quand et comment Nagios soit vérifier son bon fonctionnement,
- définit comment informer les personnes responsables de ce service.

Créez donc le fichier **/usr/local/nagios/etc/services/localhost-www.cfg** pour surveiller le serveur web de notre hôte Nagios :

```
root@nagios:~# vi /usr/local/nagios/etc/services/localhost-www.cfg
root@nagios:~# cat /usr/local/nagios/etc/services/localhost-www.cfg
  define service{
    host_name                       localhost
    service_description             www
    check_command                   check_http
    check_interval                  10
    check_period                    24x7
    retry_interval                  3
    max_check_attempts              3
    notification_interval           30
    notification_period             24x7
    notification_options            w,c,u,r
    contact_groups                  admins
    }
```

ainsi que le fichier **/usr/local/nagios/etc/services/localhost-ssh.cfg** pour surveiller le serveur ssh :

```
root@nagios:~# vi /usr/local/nagios/etc/services/localhost-ssh.cfg
root@nagios:~# cat /usr/local/nagios/etc/services/localhost-ssh.cfg
  define service{
```

```
    host_name                       localhost
    service_description             ssh
    check_command                   check_ssh
    check_interval                  5
    retry_interval                  1
    max_check_attempts              3
    check_period                    24x7
    contact_groups                  admins
    notification_interval           60
    notification_period             24x7
    notification_options            w,c,u,r
    }
```

Le tableau suivant résume les directives usuelles pour décrire un service :

Directive	**Description**
host_name	Une liste de noms courts d'hôtes, séparés par des virgules, sur lequels le service fonctionne.
hostgroup_name	Une liste de noms courts de groupes d'hôtes, séparés par des virgules, sur lequels le service fonctionne.
service_description	Un nom descriptif unique du service.
servicegroups	Une liste de tous les groupes de services, séparés par des virgules, dont le service courant devait être membre.
check_command	Le nom court de la commande qui devrait être utilisée pour vérifier si le service fonctionne.
check_interval	Spécifie en minutes la fréquence des tests.
retry_interval	Spécifie le nombre de minutes à attendre avant de procéder de nouveau à un test dans le cas où l'état est DOWN.
max_check_attempts	Spécifie le nombre de tests à éffectuer avant que Nagios considère que l'état du service est DOWN.
check_period	Spécifie le nom de la plage temporelle utilisée pour déterminer les plages d'horaires pendant lesquelles les tests sont effectués.
contacts	Une liste de contacts, séparés par des virgules, qui doivent être contactés lors d'un changement de l'état du service.
contact_groups	Une liste de groupe de contacts, séparés par des virgules, qui doivent être contactés lors d'un changement de l'état du service.
first_notification_delay	Spécifie le nombre de minutes à attendre avant que la première notification d'un changement d'état soit envoyée.
notification_interval	Spécifie la fréquence de l'envoi des notifications.
notification_period	Spécifie le nom de la plage temporelle utilisée pour déterminer les plages d'horaires pendant lesquelles les notifications seront envoyées.

Directive	Description
notification_options	Une liste de type de notification d'états, séparés par des virgules, qui déclencheront une notification. Doit être une ou plusieurs des lettres suivants : **w** - WARNING, **u** - UNKNOWN, **C** - CRITICAL, **r** - recovery (OK), **f** - FLAPPING (oscillation d'état) ou **s** - scheduled downtime (début ou fin d'un arrêt planifié.

Le même service peut être surveillé sur des hôtes différents en séparant les noms d'hôtes par une virgule :

```
define service{
  hostgroup_name                 linux-servers
  host_name                      localhost,aix01
  service_description            SSH
  (...)
  }
```

Pour exclure un hôte ou un groupe d'hôtes, il convient d'utiliser le caractère **!** :

```
define service{
  hostgroup_name                 linux-servers
  host_name                      !linuxbox01,aix01
  service_description            SSH
  (...)
  }
```

La Configuration des Groupes de Services

Les services peuvent être groupés de la même manière que les hôtes.

Par exemple :

```
define servicegroup{
  servicegroup_name     databaseservices
  alias                 All services related to databases
  members               linuxbox01,mysql,linuxbox01,
```

```
                          pgsql,aix01,db2
  }
```

Important : Ce groupe de services, identifé par son nom unique **databaseservices** consiste en mysql et pgsql sur l'hôte linuxbox01 et db2 sur l'hôte aix01.

Le tableau suivant résume les directives usuelles pour décrire un groupe de services :

Directive	Description
servicegroup_name	Le nom court et unique du groupe de services.
alias	Un nom descriptif du groupe de services.
members	Une liste d'hôtes et de services, séparés par des virgules, qui sont membre du groupe de services courant.
servicegroup_members	Une liste de groupe de services, séparés par des virgules, dont les membres doivent aussi être membres du groupe de services courant.

Il est aussi possible de definir des groupes dont un service sera membre, à l'intérieur de la définition du service lui-même :

```
define servicegroup{
  servicegroup_name     databaseservices
  alias                 All services related to databases
  }

define service{
  host_name             linuxbox01
  service_description   mysql
  check_command         check_ssh
  servicegroups         databaseservices
  }
```

La Configuration des Commandes

La définition d'une commande spécifie :

- comment auront lieu les vérifications des hôtes et des services,
- comment auront lieu les notifications de problèmes,
- comment fonctionne le gestionnaire d'événements.

Par exemple :

```
define command{
  command_name     check-host-alive
  command_line     $USER1$/check_ping -H $HOSTADDRESS$
                   -w 3000.0,80% -c 5000.0,100% -p 5
  }
```

La définition d'un hôte qui utiliserait cette commande pourrait être :

```
define host{
  host_name        somemachine
  address          10.0.0.1
  check_command    check-host-alive
  }
```

Une commande peut également recevoir des arguments :

```
define command{
  command_name     check-host-alive-limits
  command_line     $USER1$/check_ping -H $HOSTADDRESS$
                   -w $ARG1$ -c $ARG2$ -p 5
  }
```

Pour passer les valeurs des arguments à la commande la définition de l'hôte serait :

```
define host{
  host_name        othermachine
  address          10.0.0.2
  check_command    check-host-alive-limits!3000.0,80%!5000.0,100%
  }
```

La Configuration des Plages Temporelles

Une configuration d'une plage temporelle contient :

- des dates et des horaires pendant lesquels une action devrait avoir lieu,
- des dates et des horaires pendant lesquels des notifications devraient être envoyées,
- un nom unique,
- une description.

Par exemple :

```
define timeperiod{
  timeperiod_name  workinghours
  alias            Working Hours, from Monday to Friday
  monday           09:00-17:00
  tuesday          09:00-17:00
  wednesday        09:00-17:00
  thursday         09:00-17:00
  friday           09:00-17:00
  }
```

Les formats des dates sont :

- **Date Calendaire** (Calendar date) - par exemple 2019-11-01,
- **Date Recurrente Annuelle** (Date recurring every year) - par exemple **July 4** qui indique le 4 juillet de **chaque** année,
- **Jour Spécifique dans le Mois** (Specific day within a month) - par exemple **day 14** qui indique le 14ième jour de **chaque** mois,
- **Jour de la Semaine Spécifique dans un Mois avec Positionnement** (Specific weekday along with offset in a month) - par exemple **Monday**

1 September qui indique le premier lundi de **chaque** mois de septembre ou bien **Monday -1 May** qui indique le dernier lundi de **chaque** mois de mai,
- **Jour de la Semaine Spécifique dans un Mois** (Specific weekday in all months) - par exemple **Monday 1** qui indique le premier lundi de **chaque** mois de l'année,
- **Jour de la Semaine** (Weekday) - par exemple **Monday** qui indique **chaque** lundi.

Afin de pouvoir configurer correctment les objets de notre serveur Nagios, créez maintenant le fichier **/usr/local/nagios/etc/timeperiods/default.cfg** :

```
root@nagios:~# vi /usr/local/nagios/etc/timeperiods/default.cfg
root@nagios:~# cat /usr/local/nagios/etc/timeperiods/default.cfg
  define timeperiod{
    timeperiod_name      workinghours
    alias                Working Hours, from Monday to Friday
    monday               09:00-17:00
    tuesday              09:00-17:00
    wednesday            09:00-17:00
    thursday             09:00-17:00
    friday               09:00-17:00
    }

  define timeperiod{
    timeperiod_name      weekends
    alias                Weekends all day long
    saturday             00:00-24:00
    sunday               00:00-24:00
    }

  define timeperiod{
    timeperiod_name      24x7
    alias                24 hours a day 7 days a week
    monday               00:00-24:00
    tuesday              00:00-24:00
    wednesday            00:00-24:00
```

```
    thursday             00:00-24:00
    friday               00:00-24:00
    saturday             00:00-24:00
    sunday               00:00-24:00
    }
```

Il est aussi possible de spécifier des plages multiples en les séparant par des virgules :

```
  define timeperiod{
    timeperiod_name      workinghours
    alias                Working Hours, excluding lunch break
    monday               09:00-13:00,14:00-17:00
    tuesday              09:00-13:00,14:00-17:00
    wednesday            09:00-13:00,14:00-17:00
    thursday             09:00-13:00,14:00-17:00
    friday               09:00-13:00,14:00-17:00
    }
```

ou bien d'exclure une plage spécifique quand une autre plage est active :

```
  define timeperiod{
    timeperiod_name      first-mondays
    alias                First Mondays of each month
    monday 1 january     00:00-24:00
    monday 1 february    00:00-24:00
    monday 1 march       00:00-24:00
    monday 1 april       00:00-24:00
    monday 1 may         00:00-24:00
    monday 1 june        00:00-24:00
    monday 1 july        00:00-24:00
    monday 1 august      00:00-24:00
    monday 1 september   00:00-24:00
    monday 1 october     00:00-24:00
    monday 1 november    00:00-24:00
```

```
    monday 1 december    00:00-24:00
  define timeperiod{
    timeperiod_name      workinghours-without-first-monday
    alias                Working Hours, without first Monday of
                         each month
    monday               09:00-17:00
    tuesday              09:00-17:00
    wednesday            09:00-17:00
    thursday             09:00-17:00
    friday               09:00-17:00
    exclude              first-mondays
    }
```

La Configuration des Contacts

La définition d'un contact contient :

- un nom court unique,
- un nom déscriptif,
- un moyen de contact tel une adresse email ou le numéro d'un beeper.

Par exemple :

```
define contact{
    contact_name                    jdoe
    alias                           John Doe
    email                           john.doe@yourcompany.com
    contactgroups                   admins,nagiosadmin
    host_notification_period        workinghours
    service_notification_period     workinghours
    host_notification_options       d,u,r
    service_notification_options    w,u,c,r
    host_notification_commands      notify-host-by-email
```

```
    service_notification_commands  notify-service-by-email
    }
```

Créez maintenant le fichier **/usr/local/nagios/etc/contacts/trainee.cfg** :

```
root@nagios:~# vi /usr/local/nagios/etc/contacts/trainee.cfg
root@nagios:~# cat /usr/local/nagios/etc/contacts/trainee.cfg
define contact{
    contact_name                    trainee
    alias                           Hugh Norris
    email                           infos@i2tch.eu
    contactgroups                   admins,nagiosadmin
    host_notification_period        workinghours
    service_notification_period     workinghours
    host_notification_options       d,u,r
    service_notification_options    w,u,c,r
    host_notification_commands      notify-host-by-email
    service_notification_commands   notify-service-by-email
    }
```

Important : **Remplacez** les valeurs des options **alias** et **email** ci-dessus avec **vos propres coordonnées**.

Le tableau suivant résume les directives usuelles pour décrire un contact :

Directive	**Description**
contact_name	Le nom court et unique du contact.
alias	Un nom descriptif du contact, normallement prénom nom.
host_notifications_enabled	Spécifie si le contact courant doit recevoir les notifications concernant les états des hôtes.
host_notification_period	Spécifie le nom de la plage temporelle utilisée pour déterminer les plages d'horaires pendant lesquelles les notifications concernant l'état des hôtes seront reçues par le contact courant.

Directive	Description
host_notification_commands	Spécifie une ou plusieurs commandes, séparées par des virgules, qui doit ou doivent être utilisée(s) pour notifier le contact courant d'un changement d'état.
host_notification_options	Une liste de type de notification d'états, séparés par des virgules, dont le contact courant sera notifié. Doit être un ou plusieurs des lettres suivants : **d** - DOWN, **u** - UNREACHABLE, **r** - recovery (UP), **f** - FLAPPING (oscillation d'état), **s** - scheduled downtime (début ou fin d'un arrêt planifié ou **n** - aucune notification.
service_notifications_enabled	Spécifie si le contact courant doit recevoir les notifications concernant les états des services.
service_notification_period	Spécifie le nom de la plage temporelle utilisée pour déterminer les plages d'horaires pendant lesquelles les notifications concernant l'état des services seront reçues par le contact courant.
service_notification_commands	Spécifie une ou plusieurs commandes, séparées par des virgules, qui doit ou doivent être utilisée(s) pour notifier le contact courant d'un changement d'état d'un service.
service_notification_options	Une liste de type de notification d'états, séparés par des virgules, dont le contact courant sera notifié. Doit être une ou plusieurs des lettres suivants : **w** - WARNING, **u** - UNKNOWN, **C** - CRITICAL, **r** - recovery (OK), **f** - FLAPPING (oscillation d'état), **s** - scheduled downtime (début ou fin d'un arrêt planifié) ou **n** - aucune notification.
email	L'adresse email du contact.
pager	Le numéro du beeper du contact ou une adresse email vers la passerelle du service.
adresse1 à adresse6	Six champs supplémentaires pour des information de contact.
can_submit_commands	Spécifie si oui ou non le contact courant peut exécuter des commandes à partir de l'interface HTML de Nagios.
retain_status_information	Spécifie si oui ou non les informations de statut du contact courant seront disponibles entre démarrages.
retain_nonstatus_information	Spécifie si oui ou non les informations autres que celles de statut du contact courant seront disponibles entre démarrages.

La Configuration des Groupes de Contacts

Comme les autres objets, les contacts peuevnt être groupés.

Par exemple :

```
  define contactgroup{
    contactgroup_name             linux-admins
    alias                         Linux Administrators
    members                       jdoe,asmith
    }
```

Le tableau suivant résume les directives usuelles pour décrire un groupe de contacts :

Directive	**Description**
contactgroup_name	Le nom court et unique du groupe de contacts.
alias	Un nom descriptif du groupe de contacts.
members	Une liste de contacts, séparés par des virgules, qui sont membre du groupe de contacts courant.
contactgroup_members	Une liste de groupe de contacts, séparés par des virgules, dont les membres doivent aussi être membres du groupe de contacts courant.

Afin que notre utilisateur trainee puisse fonctionner correctement, créez maintenant le fichier **/usr/local/nagios/etc/contactgroups/admins.cfg** :

```
root@nagios:~# vi /usr/local/nagios/etc/contactgroups/admins.cfg
root@nagios:~# cat /usr/local/nagios/etc/contactgroups/admins.cfg
  define contactgroup{
    contactgroup_name             admins
    alias                         System administrators
  }

  define contactgroup{
    contactgroup_name             nagiosadmin
    alias                         Nagios administrators
    }
```

Vérification de la Nouvelle Configuration

Avant de vérifier la configuration, il convient de faire une sauvegarde du fichier **/usr/local/nagios/etc/nagios.cfg** :

```
root@nagios:~# cp /usr/local/nagios/etc/nagios.cfg /root
```

Copiez maintenant le fichier **/tmp/nagios.cfg** vers le répertoire **/usr/local/nagios/etc/** :

```
root@nagios:~# cp /tmp/nagios.cfg /usr/local/nagios/etc
```

Arrêtez maintenant le servur Nagios :

```
root@nagios:~# systemctl stop nagios.service
```

Vérifiez maintenant votre configuration :

```
root@nagios:~# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.2.0
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 08-01-2016
License: GPL

Website: https://www.nagios.org
Reading configuration data...
   Read main config file okay...
   Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
    Checked 2 services.
    Checked 1 hosts.
    Checked 0 host groups.
    Checked 0 service groups.
    Checked 1 contacts.
    Checked 2 contact groups.
    Checked 24 commands.
    Checked 3 time periods.
    Checked 0 host escalations.
    Checked 0 service escalations.
Checking for circular paths...
    Checked 1 hosts
```

```
    Checked 0 service dependencies
    Checked 0 host dependencies
    Checked 3 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors:   0

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

Les Notifications

Les notifications peuvent être envoyées dans n'importe laquelle des conditions suivantes :

Condition	**Notification**
Hôte devient DOWN ou UNREACHABLE	Envoyée après first_notification_delay.
Hôte reste dans l'état DOWN ou UNREACHABLE	Envoyée tous les notification_interval.
Hôte regagne l'état UP	Envoyée immédiatement et seulement une fois.
Hôte démarre ou arrête une oscillation d'état (Flapping)	Envoyée immédiatement.
Hôte continue une oscillation d'état (Flapping)	Envoyée tous les notification_interval.
Service devient WARNING, CRITICAL ou UNKNOWN	Envoyée après first_notification_delay.
Service reste dans l'état WARNING, CRITICAL ou UNKNOWN	Envoyée tous les notification_interval.
Service regagne l'état OK	Envoyée immédiatement et seulement une fois.
Service démarre ou arrête une oscillation d'état (Flapping)	Envoyée immédiatement.
Service continue une oscillation d'état (Flapping)	Envoyée tous les notification_interval.

Si une des conditions est remplie, Nagios va commencer à évaluer si une notification doit être envoyée et dans ce cas à qui :

1. Pour commencer, Nagios vérifie que l'heure et la date actuelle soient référencées par la **notification_timeperiod** dans l'hôte ou le service concerné,

2. Ensuite Nagios construit une liste d'utilisateurs basée sur les valeurs de **contacts** et **contact_groups**,
3. Pour chaque utilisateur trouvé, Nagios vérifie la valeur de **host_notification_period** et de **service_notification_period**,
4. Dans le cas d'un hôte, la valeur de **host_notification_options** est lue tandis que dans le cas d'un service, la valeur de **service_notification_options** est lue,
5. Si aucune des vérifications précédentes est en échec, Nagios envarra un message au(x) contact(s) selon les valeurs de **host_notification_commands** ou de **service_notification_commands**.

Héritage et Modèles

Nagios permet l'utilisation de **modèles** ausii appellés des **gabarits** afin de fournir un système d'héritage :

- Les modèles sont des objets Nagios ayant la directive **register** fixée à **0**,
- Le nom d'un modèle est spécifié par la directive **name** à l'opposé d'un hôte qui utilise la directive **host_name** et un service qui est référencé par la directive **service_description**,
- Un objet qui hérite ses paramètres d'un modèle doit référencé ce dernier en utilisant la directive **use**.

Par exemple :

```
define host{
  name                            generic-server
  check_command                   check-host-alive
  check_interval                  5
  retry_interval                  1
  max_check_attempts              5
  check_period                    24x7
  notification_interval           30
  notification_period             24x7
  notification_options            d,u,r
  register                        0
}

define host{
  use                             generic-server
```

```
  host_name                       linuxbox01
  alias                           Linux Server 01
  address                         10.0.2.1
  contact_groups                  linux-admins
  }
```

Il est aussi possible d'hériter des paramètres de plusieurs modèles :

```
define service{
  name                          generic-service
  check_interval                10
  retry_interval                2
  max_check_attempts            3
  check_period                  24x7
  register                      0
}

define service{
  host_name                     workinghours-service
  check_period                  workinghours
  notification_interval         30
  notification_period           workinghours
  notification_options          w,c,u,r
  register                      0
}

define service{
  use                           workinghours-service,generic-service
  contact_groups                linux-admins
  host_name                     linuxbox01
  service_description           SSH
  check_command                 check_ssh
  }
```

Important : Notez dans l'exemple ci-dessus la duplicité de la directive **check-period**. L'objet dont le **service_description** est **SSH** utilisera la valeur **workinghours** parce que la directive **use** spécifie le modèle **workinghours-service** en premier. Pour utiliser la valeur de la directive **check_period** du modèle **generic_service**, il convient simplement d'inverser l'ordre dans la directive **use** - **generic-service,workinghours-service**.

Un modèle peut aussi héritier des paramètres d'un autre modèle :

```
define host{
    host_name       linuxserver1
    use             generic-linux,template-chicago
    .....
    }
define host{
    register        0
    name            generic-linux
    use             generic-server
    .....
    }
define host{
    register        0
    name            generic-server
    use             generic-host
    .....
    }
define host{
    register        0
    name            template-chicago
    use             contacts-chicago,misc-chicago
    .....
    }
```

Important : Dans l'exemple ci-dessus, Nagios cherche d'abord des paramètres dans l'objet **linuxserver1** puis dans les autres objets dans l'ordre suivant : **generic-linux**, **generic-server**, **generic-host**, **template-chicago**, **contacts-chicago** et **misc-chicago**.

LAB #1 - Mise en Place d'un Client Linux à Surveiller

Préparation

Insérez le CDROM des Additions VirtualBox dans votre VM :

```
Périphériques > Insérer l'image CD des Additions Invité ...
```

Installez ensuite les dépendances necessaires :

```
root@nagios:~# apt-get update
root@nagios:~# apt-get install dkms build-essential module-assistant
```

Configurez le système pour la construction de modules :

```
root@nagios:~# m-a prepare
```

Installez les Additions :

```
root@nagios:~# sh /media/cdrom/VBoxLinuxAdditions.run
```

Re-démarrez votre VM.

A Faire : Arrêtez le serveur Nagios. Dans l'interface de VirtualBox, mettez la VM Debian 8 dans le Réseau NAT **NatNetwork**. Démarrez ensuite la VM Debian 8.

Mise en Place

Lancez maintenant une VM CentOS vierge.

Installez ensuite le dépot **epel** :

```
[root@centos7 ~]# yum install epel-release
```

Installez maintenant les paquets necéssaires :

```
[root@centos7 ~]# yum install nrpe nagios-plugins-all openssl
```

Modifiez la section **allowed_hosts** du fichier **/etc/nagios/nrpe.cfg** en ajoutant l'adresse IP du serveur Nagios :

```
[root@centos7 ~]# vi /etc/nagios/nrpe.cfg
[root@centos7 ~]# cat vi /etc/nagios/nrpe.cfg
...
# ALLOWED HOST ADDRESSES
# This is an optional comma-delimited list of IP address or hostnames
# that are allowed to talk to the NRPE daemon. Network addresses with a bit mask
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currently
# supported.
#
# Note: The daemon only does rudimentary checking of the client's IP
# address.  I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
```

```
# NOTE: This option is ignored if NRPE is running under either inetd
# or xinetd or systemd

allowed_hosts=127.0.0.1,::1,10.0.2.14
...
```

Configurez une adresse IP fixe :

```
[root@centos7 ~]# nmcli c show
NAME                UUID                                  TYPE            DEVICE
Wired connection 1  ab5d9069-3caf-34cb-b92f-206228ed53bf  802-3-ethernet  enp0s3
[root@centos7 ~]# nmcli connection add con-name ip_fixe ifname enp0s3 type ethernet ip4 10.0.2.16/24 gw4 10.0.2.2
Connection 'ip_fixe' (ac72911c-fcce-4778-bddd-66571da93826) successfully added.
[root@centos7 ~]# nmcli c show
NAME                UUID                                  TYPE            DEVICE
Wired connection 1  ab5d9069-3caf-34cb-b92f-206228ed53bf  802-3-ethernet  enp0s3
ip_fixe             ac72911c-fcce-4778-bddd-66571da93826  802-3-ethernet  --
[root@centos7 ~]# nmcli connection up ip_fixe
```

Déconnectez-vous et reconnectez-vous à votre VM CentOS 7 en utilisant SSH.

```
[root@centos7 ~]# nmcli c show
NAME                UUID                                  TYPE            DEVICE
ip_fixe             ac72911c-fcce-4778-bddd-66571da93826  802-3-ethernet  enp0s3
Wired connection 1  ab5d9069-3caf-34cb-b92f-206228ed53bf  802-3-ethernet  --
[root@centos7 ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:00:27:03:97:dd brd ff:ff:ff:ff:ff:ff
```

```
    inet 10.0.2.16/24 brd 10.0.2.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::ec96:c083:d447:7ff9/64 scope link
       valid_lft forever preferred_lft forever
```

Démarrez le service **nrpe** et vérifiez qu'il permet des connexions à partir de l'adresse du serveur Nagios (**Allowing connections from: 127.0.0.1 10.0.2.14**) :

```
[root@centos7 ~]# systemctl enable nrpe
Created symlink from /etc/systemd/system/multi-user.target.wants/nrpe.service to
/usr/lib/systemd/system/nrpe.service.
[root@centos7 ~]# systemctl start nrpe
[root@centos7 ~]# systemctl status nrpe
● nrpe.service - Nagios Remote Program Executor
   Loaded: loaded (/usr/lib/systemd/system/nrpe.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2018-05-15 11:43:01 CEST; 6s ago
     Docs: http://www.nagios.org/documentation
  Process: 21053 ExecStart=/usr/sbin/nrpe -c /etc/nagios/nrpe.cfg -d $NRPE_SSL_OPT (code=exited,
status=0/SUCCESS)
 Main PID: 21054 (nrpe)
   CGroup: /system.slice/nrpe.service
           └─21054 /usr/sbin/nrpe -c /etc/nagios/nrpe.cfg -d

May 15 11:43:01 centos7.fenestros.loc systemd[1]: Starting Nagios Remote Program Executor...
May 15 11:43:01 centos7.fenestros.loc nrpe[21054]: Starting up daemon
May 15 11:43:01 centos7.fenestros.loc nrpe[21054]: Server listening on 0.0.0.0 port 5666.
May 15 11:43:01 centos7.fenestros.loc nrpe[21054]: Server listening on :: port 5666.
May 15 11:43:01 centos7.fenestros.loc nrpe[21054]: Listening for connections on port 5666
May 15 11:43:01 centos7.fenestros.loc nrpe[21054]: Allowing connections from: 127.0.0.1,::1,10.0.2.14
May 15 11:43:01 centos7.fenestros.loc systemd[1]: Started Nagios Remote Program Executor.
```

Dernièrement, modifiez le nom d'hôte du client :

```
[root@centos7 ~]# nmcli g hostname linux.i2tch.loc
```

A Faire : Arrêtez la VM CentOS 7. Dans l'interface de VirtualBox, mettez la VM CentOS 7 dans le Réseau NAT **NatNetwork**. Démarrez ensuite la VM CentOS 7.

Configurer le Serveur Nagios pour la Surveillance

Afin de surveiller la VM CentOS 7, créez maintenant un hôte dans le répertoire **/usr/local/nagios/etc/hosts** appelé **linux.cfg** :

```
root@nagios:~# vi /usr/local/nagios/etc/hosts/linux.cfg
root@nagios:~# cat /usr/local/nagios/etc/hosts/linux.cfg
  define host{
    host_name                       linux
    alias                           Linux
    address                         10.0.2.16
    check_command                   check-host-alive
    check_interval                  5
    retry_interval                  1
    max_check_attempts              5
    check_period                    24x7
    contact_groups                  admins
    notification_interval           60
    notification_period             24x7
    notification_options            d,u,r
    }
```

Créez ensuite le fichier **/usr/local/nagios/etc/services/linux-ssh.cfg** pour surveiller le serveur ssh :

```
root@nagios:~# vi /usr/local/nagios/etc/services/linux-ssh.cfg
root@nagios:~# cat /usr/local/nagios/etc/services/linux-ssh.cfg
  define service{
```

```
    host_name                       linux
    service_description             ssh
    check_command                   check_ssh
    check_interval                  5
    retry_interval                  1
    max_check_attempts              3
    check_period                    24x7
    contact_groups                  admins
    notification_interval           60
    notification_period             24x7
    notification_options            w,c,u,r
    }
```

Re-démarrez le service nagios :

```
[root@server ~]# systemctl restart nagios
```

Utilisation de l'Interface HTML de Nagios

L'Interface HTML de Nagios

Les fichiers HTML se trouvent dans le répertoire **/usr/local/nagios/share/** :

```
root@nagios:~# ls -l /usr/local/nagios/share
total 192
drwxrwsr-x 4 nagios nagios 4096 Apr 24 12:56 angularjs
drwxrwsr-x 3 nagios nagios 4096 Apr 24 12:56 bootstrap-3.3.0
-rw-rw-r-- 1 nagios nagios  576 Apr 24 12:56 config.inc.php
drwxrwsr-x 2 nagios nagios 4096 Apr 24 12:56 contexthelp
drwxrwsr-x 2 nagios nagios 4096 Apr 24 12:56 d3
drwxrwsr-x 3 nagios nagios 4096 Apr 24 12:56 docs
```

```
-rw-rw-r-- 1 nagios nagios  879 Apr 24 12:56 graph-header.html
-rw-rw-r-- 1 nagios nagios 4136 Apr 24 12:56 histogram-form.html
-rw-rw-r-- 1 nagios nagios 5579 Apr 24 12:56 histogram-graph.html
-rw-rw-r-- 1 nagios nagios 2855 Apr 24 12:56 histogram.html
-rw-rw-r-- 1 nagios nagios 2372 Apr 24 12:56 histogram-links.html
drwxrwsr-x 3 nagios nagios 4096 Apr 24 12:56 images
drwxrwsr-x 3 nagios nagios 4096 Apr 24 12:56 includes
-rw-rw-r-- 1 nagios nagios 1931 Apr 24 12:56 index.php
-rw-rw-r-- 1 nagios nagios 1660 Apr 24 12:56 infobox.html
drwxrwsr-x 2 nagios nagios 4096 Apr 24 12:56 js
-rw-rw-r-- 1 nagios nagios 1177 Apr 24 12:56 jsonquery.html
drwxr-sr-x 4 root   nagios 4096 Apr 24 14:32 locale
-rw-rw-r-- 1 nagios nagios 8745 Apr 24 12:56 main.php
-rw-rw-r-- 1 nagios nagios 2015 Apr 24 12:56 map-directive.html
-rw-rw-r-- 1 nagios nagios 3824 Apr 24 12:56 map-form.html
-rw-rw-r-- 1 nagios nagios  647 Apr 24 12:56 map-links.html
-rw-rw-r-- 1 nagios nagios 3298 Apr 24 12:56 map.php
-rw-rw-r-- 1 nagios nagios 1723 Apr 24 12:56 map-popup.html
drwxrwsr-x 2 nagios nagios 4096 Apr 24 12:56 media
-rw-rw-r-- 1 nagios nagios   26 Apr 24 12:56 robots.txt
-rw-rw-r-- 1 nagios nagios  657 Apr 24 12:56 rss-corebanner.php
-rw-rw-r-- 1 nagios nagios 1071 Apr 24 12:56 rss-corefeed.html
-rw-rw-r-- 1 nagios nagios  898 Apr 24 12:56 rss-corefeed.php
-rw-rw-r-- 1 nagios nagios  544 Apr 24 12:56 rss-newsfeed.html
-rw-rw-r-- 1 nagios nagios 1068 Apr 24 12:56 rss-newsfeed.php
-rw-rw-r-- 1 nagios nagios 6163 Apr 24 12:56 side.php
drwxrwsr-x 2 nagios nagios 4096 Apr 24 12:56 spin
drwxrwsr-x 2 nagios nagios 4096 Apr 24 12:56 ssi
drwxrwsr-x 2 nagios nagios 4096 Apr 24 12:56 stylesheets
-rw-rw-r-- 1 nagios nagios 4243 Apr 24 12:56 trends-form.html
-rw-rw-r-- 1 nagios nagios 4256 Apr 24 12:56 trends-graph.html
-rw-rw-r-- 1 nagios nagios  852 Apr 24 12:56 trends-host-yaxis.html
-rw-rw-r-- 1 nagios nagios 2990 Apr 24 12:56 trends.html
-rw-rw-r-- 1 nagios nagios 3586 Apr 24 12:56 trends-links.html
```

```
-rw-rw-r-- 1 nagios nagios 1065 Apr 24 12:56 trends-service-yaxis.html
```

Les binaires CGI se trouvent dans le répertoire **/usr/local/nagios/sbin/** :

```
root@nagios:~# ls -l /usr/local/nagios/sbin
total 5116
-rwxrwxr-x 1 nagios nagios 321144 Apr 24 12:56 archivejson.cgi
-rwxrwxr-x 1 nagios nagios 302392 Apr 24 12:56 avail.cgi
-rwxrwxr-x 1 nagios nagios 295840 Apr 24 12:56 cmd.cgi
-rwxrwxr-x 1 nagios nagios 269544 Apr 24 12:56 config.cgi
-rwxrwxr-x 1 nagios nagios 310552 Apr 24 12:56 extinfo.cgi
-rwxrwxr-x 1 nagios nagios 263104 Apr 24 12:56 histogram.cgi
-rwxrwxr-x 1 nagios nagios 241864 Apr 24 12:56 history.cgi
-rwxrwxr-x 1 nagios nagios 240904 Apr 24 12:56 notifications.cgi
-rwxrwxr-x 1 nagios nagios 322936 Apr 24 12:56 objectjson.cgi
-rwxrwxr-x 1 nagios nagios 236776 Apr 24 12:56 outages.cgi
-rwxrwxr-x 1 nagios nagios 236792 Apr 24 12:56 showlog.cgi
-rwxrwxr-x 1 nagios nagios 310576 Apr 24 12:56 status.cgi
-rwxrwxr-x 1 nagios nagios 316984 Apr 24 12:56 statusjson.cgi
-rwxrwxr-x 1 nagios nagios 260576 Apr 24 12:56 statusmap.cgi
-rwxrwxr-x 1 nagios nagios 257336 Apr 24 12:56 statuswml.cgi
-rwxrwxr-x 1 nagios nagios 240936 Apr 24 12:56 statuswrl.cgi
-rwxrwxr-x 1 nagios nagios 261440 Apr 24 12:56 summary.cgi
-rwxrwxr-x 1 nagios nagios 251712 Apr 24 12:56 tac.cgi
-rwxrwxr-x 1 nagios nagios 268416 Apr 24 12:56 trends.cgi
```

Si le service Nagios est actuellement arrêté, démarrez-le :

```
root@nagios:~# systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
   Loaded: loaded (/etc/init.d/nagios)
   Active: inactive (dead) since Sat 2018-05-12 23:43:35 BST; 1h 27min ago
  Process: 20495 ExecStop=/etc/init.d/nagios stop (code=exited, status=0/SUCCESS)
  Process: 528 ExecStart=/etc/init.d/nagios start (code=exited, status=0/SUCCESS)
```

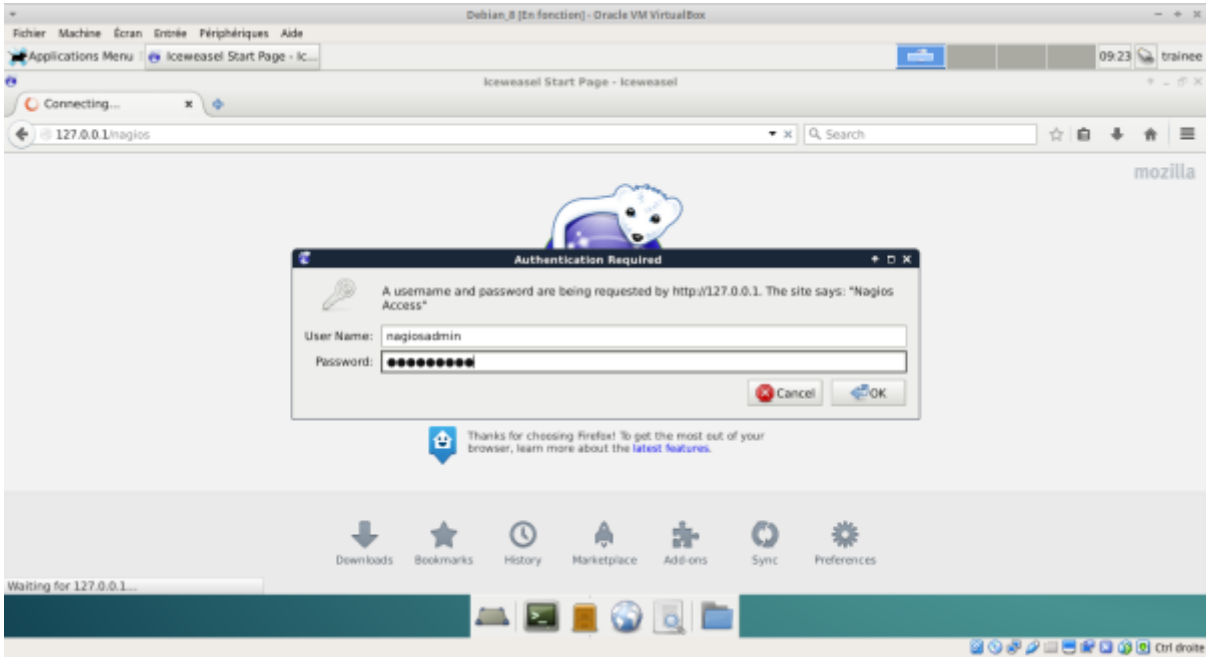
```
May 12 20:40:04 nagios.i2tch.loc nagios[725]: Auto-save of retention data completed successfully.
May 12 21:40:04 nagios.i2tch.loc nagios[725]: Auto-save of retention data completed successfully.
May 12 22:40:04 nagios.i2tch.loc nagios[725]: Auto-save of retention data completed successfully.
May 12 23:40:04 nagios.i2tch.loc nagios[725]: Auto-save of retention data completed successfully.
May 12 23:43:34 nagios.i2tch.loc systemd[1]: Stopping LSB: Starts and stops the Nagios monitoring server...
May 12 23:43:34 nagios.i2tch.loc nagios[725]: Caught SIGTERM, shutting down...
May 12 23:43:34 nagios.i2tch.loc nagios[725]: Successfully shutdown... (PID=725)
May 12 23:43:34 nagios.i2tch.loc nagios[725]: Event broker module 'NERD' deinitialized successfully.
May 12 23:43:35 nagios.i2tch.loc nagios[20495]: Stopping nagios:. done.
May 12 23:43:35 nagios.i2tch.loc systemd[1]: Stopped LSB: Starts and stops the Nagios monitoring server.
root@nagios:~# systemctl start nagios
root@nagios:~# systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
   Loaded: loaded (/etc/init.d/nagios)
   Active: active (running) since Sun 2018-05-13 01:11:24 BST; 3s ago
  Process: 20495 ExecStop=/etc/init.d/nagios stop (code=exited, status=0/SUCCESS)
  Process: 7389 ExecStart=/etc/init.d/nagios start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/nagios.service
           ├─7411 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─7413 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─7414 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─7415 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─7416 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─7417 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─7418 /usr/local/nagios/libexec/check_ping -H 127.0.0.1 -w 3000.0,80% -c 5000.0,100% -p 5
           └─7424 /bin/ping -n -U -w 30 -c 5 127.0.0.1

May 13 01:11:24 nagios.i2tch.loc nagios[7411]: nerd: Channel hostchecks registered successfully
May 13 01:11:24 nagios.i2tch.loc nagios[7411]: nerd: Channel servicechecks registered successfully
May 13 01:11:24 nagios.i2tch.loc nagios[7411]: nerd: Channel opathchecks registered successfully
May 13 01:11:24 nagios.i2tch.loc nagios[7411]: nerd: Fully initialized and ready to rock!
May 13 01:11:24 nagios.i2tch.loc nagios[7411]: wproc: Successfully registered manager as @wproc with query
handler
May 13 01:11:24 nagios.i2tch.loc nagios[7411]: wproc: Registry request: name=Core Worker 7414;pid=7414
```

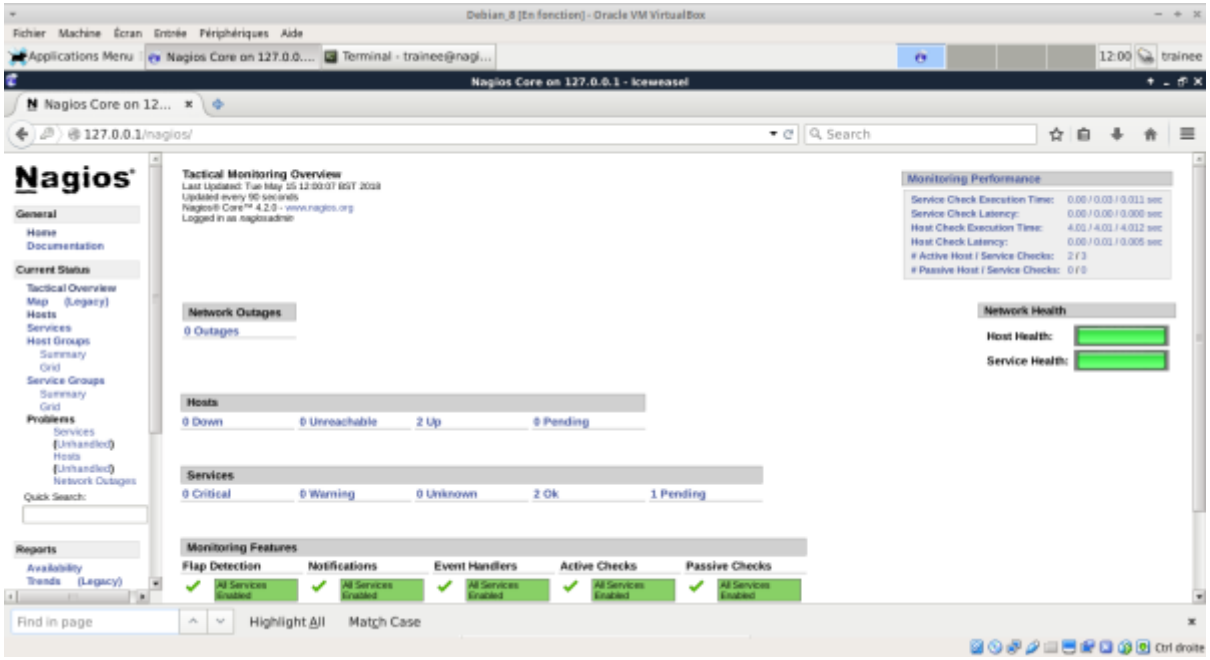
```
May 13 01:11:24 nagios.i2tch.loc nagios[7411]: wproc: Registry request: name=Core Worker 7413;pid=7413
May 13 01:11:24 nagios.i2tch.loc nagios[7411]: wproc: Registry request: name=Core Worker 7416;pid=7416
May 13 01:11:24 nagios.i2tch.loc nagios[7411]: wproc: Registry request: name=Core Worker 7415;pid=7415
May 13 01:11:24 nagios.i2tch.loc nagios[7411]: Successfully launched command file worker with pid 7417
```

Utilisation

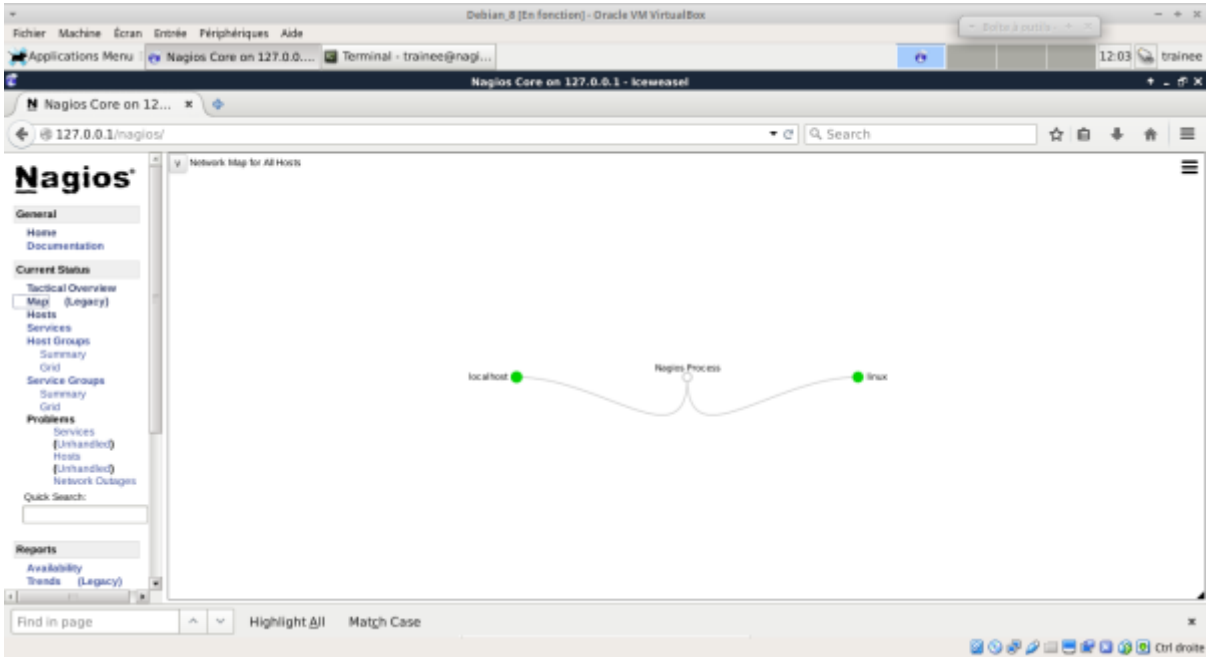
A partir du serveur Nagios en mode graphique, ouvrez le navigateur Iceweasel et saisissez l'URL **http://127.0.0.1/nagios**. Connectez vous avec les coordonnées nagiosadmin/fenestros :



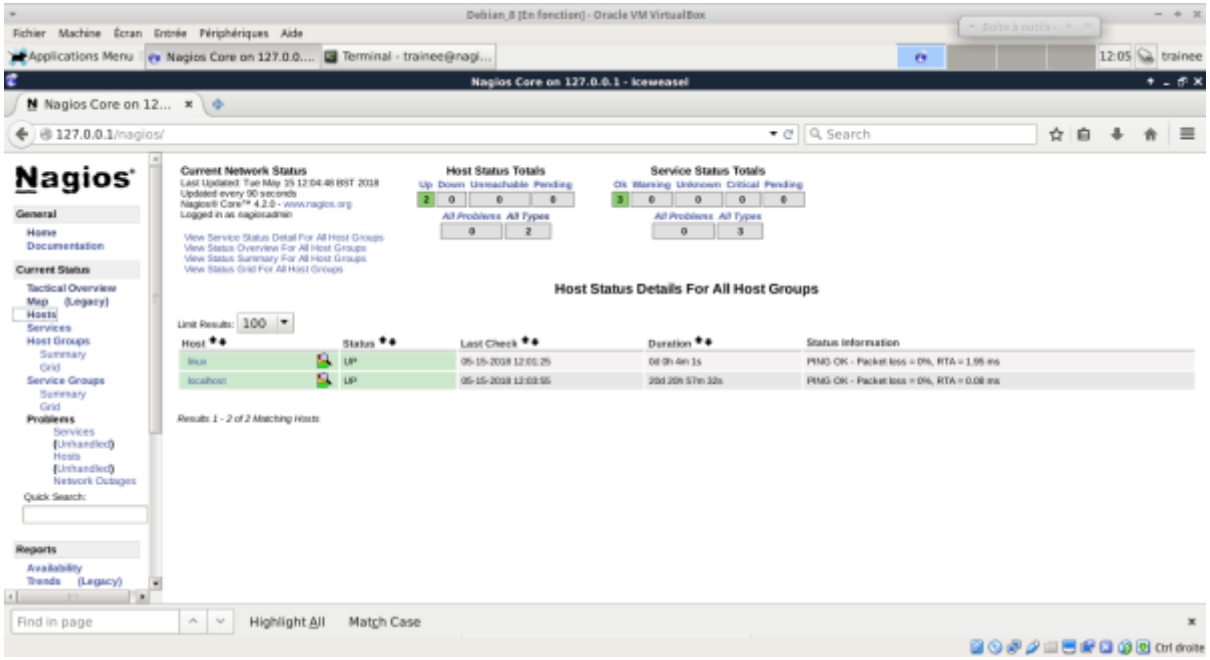
Cliquez sur le lien **Tactical Overview** dans le panneau de gauche. Cette page présente une vue globale des hôtes ainsi que les services. En haut à droite se trouve une section concernant les performances :



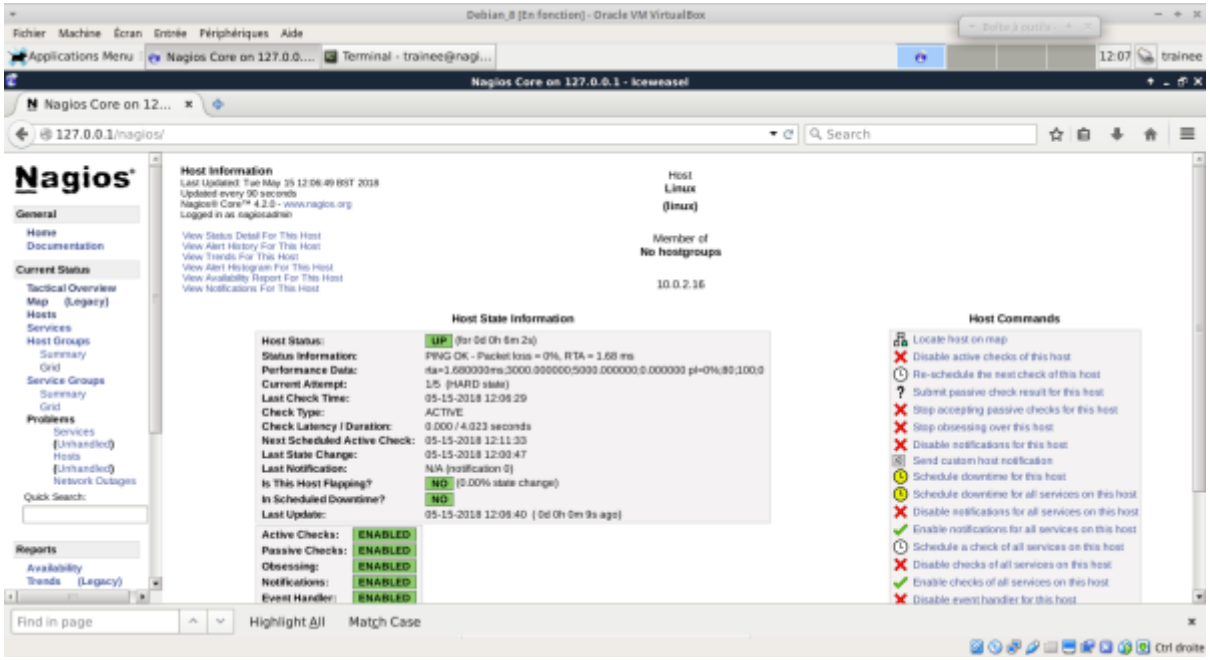
Le lien **Map** dans le panneau de gauche démontre une cartographie du réseau. Cette vbue est particulièrement utile pour visualiser les relations parent-enfant :



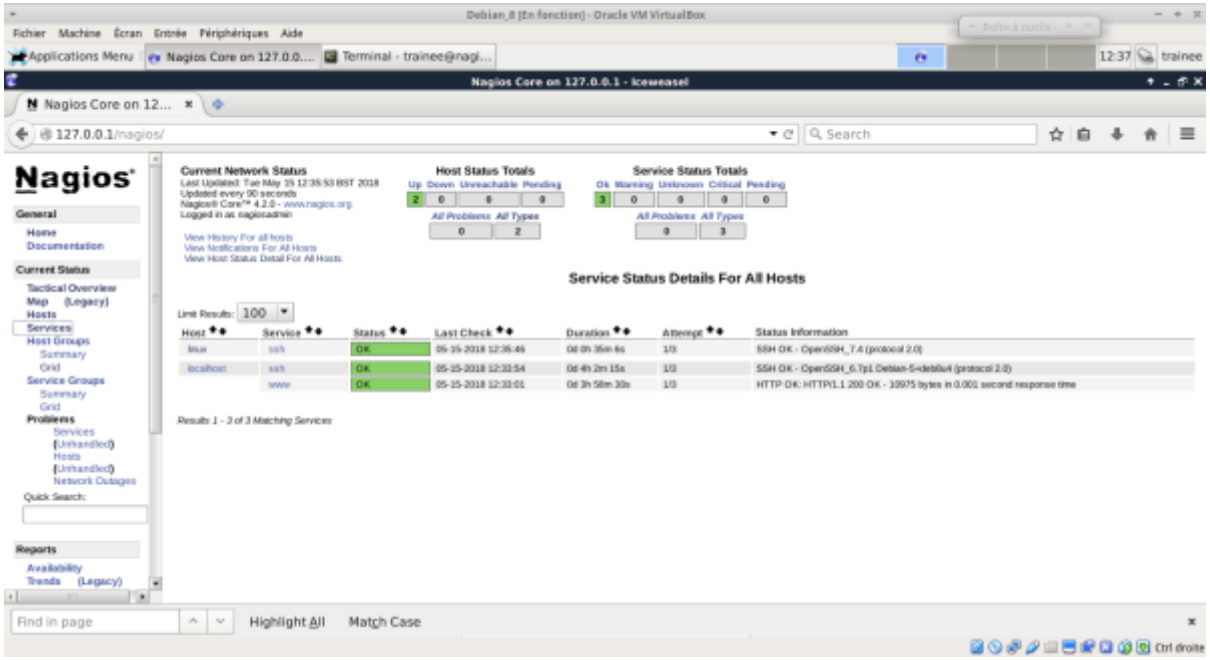
En cliquant sur le lien **Hosts** vous obtiendrez une vue permettant de voir l'ensemble des hôtes configurés :



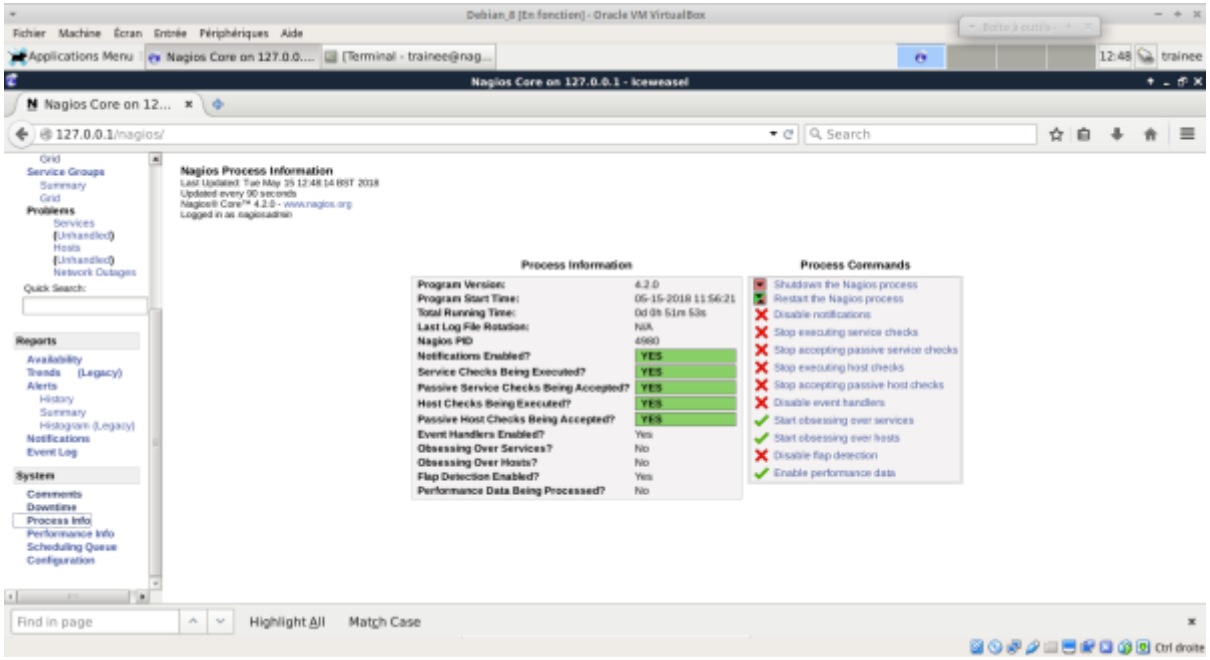
Cliquez sur le lien **linux**. Vous verrez ensuite les détails concernant cet hôte. Le menu de droite permet la configuration de Nagios pour cet hôte :



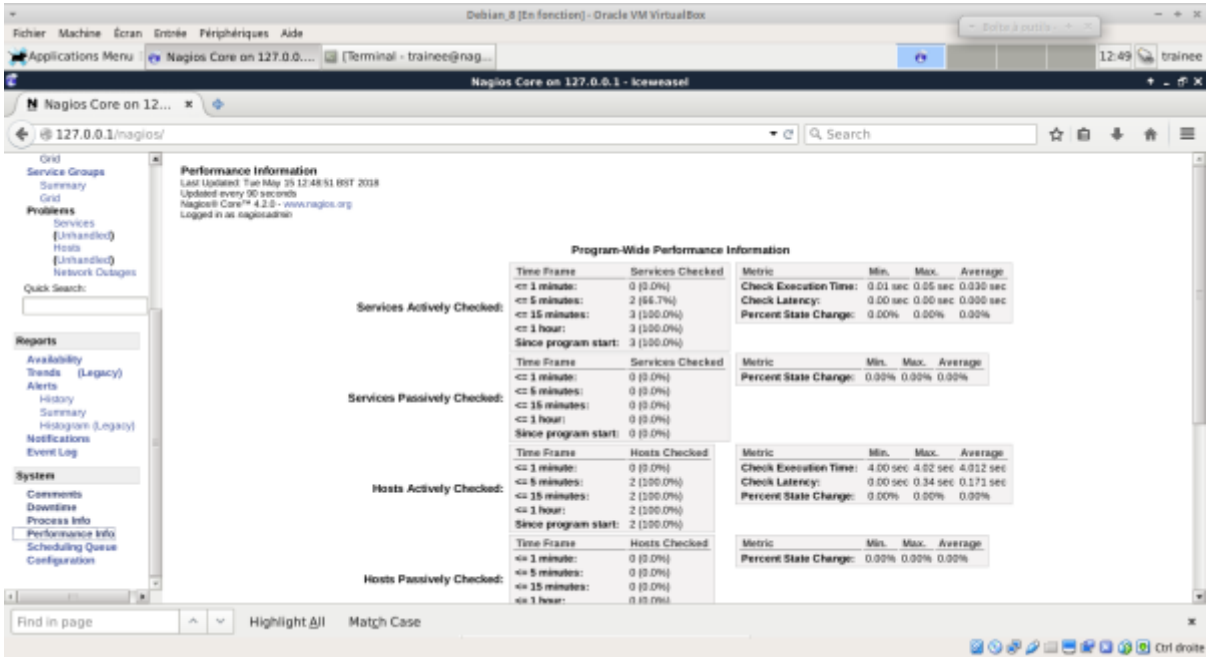
Cliquez ensuite sur le lien **Services** dans le panneau de gauche afin de visualiser l'ensemble des services surveillés :



Cliquez sur le lien **Process Info** dans le panneau de gauche afin de visualiser les informations sur le processus Nagios. Vous pouvez constater des informations sur la version de Nagios, le PID, le statut, et la rotation des journaux. Le menu de droite permet d'arrêter et de re-démarrer Nagios :



En cliquant sur le lien **Performance Info**, il est possible de consulter des informations concernant le nombre de vérifications d'hôtes et de services, le nombre de rapports reçus d'applications externes ainsi que le nombre de commandes reçues du serveur web. Cette page contient aussi des informations sur la latence. Si la latence est supérieure à 60 secondes, Nagios n'est pas capable de procéder à toutes les vérifications planifiées :



LAB #2 - Outils en Ligne de Commande

nagios_commander

Nagios Commander est un script shell qui :

- communique avec l'interface web de Nagios en utilisant une authentification HTTP,
- peut être utilisé à partir d'un poste distant du serveur Nagios,
- peut gérer plusieurs serveurs Nagios distants à partir d'une seule machine de gestion.

Pour vérifier que les dépendances sont installées, lancez la commande suivante :

```
root@nagios:~# apt-get -y install bsdmainutils curl
Reading package lists... Done
Building dependency tree
```

```
Reading state information... Done
bsdmainutils is already the newest version.
curl is already the newest version.
The following packages were automatically installed and are no longer required:
  libgconf2-4 libgnome2-0
Use 'apt-get autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 8 not upgraded.
```

Téléchargez le script nagios_commander :

```
root@nagios:~# curl -sSL https://raw.github.com/brandoconnor/nagios_commander/master/nagios_commander.sh >
/usr/local/bin/nagios_commander
root@nagios:~# ls -l /usr/local/bin
total 20
-rw-r--r-- 1 root staff 19866 May 15 13:01 nagios_commander
```

Rendez le script exécutable :

```
root@nagios:~# chmod 0755 /usr/local/bin/nagios_commander
```

Pour lister le statuts des hôtes, utilisez la commande suivante :

```
root@nagios:~# nagios_commander -n 127.0.0.1/nagios -u nagiosadmin -p fenestros -q list -h
Hostname    Status
linux      UP
localhost  UP
```

Pour faciliter l'utilisation de cette commande dans le futur, créez un alias :

```
root@nagios:~# alias ncmd='nagios_commander -n 127.0.0.1/nagios -u nagiosadmin -p fenestros'
root@nagios:~# vi .bashrc
root@nagios:~# cat .bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.
```

```
# Note: PS1 and umask are already set in /etc/profile. You should not
# need this unless you want different defaults for root.
# PS1='${debian_chroot:+($debian_chroot)}\h:\w\$ '
# umask 022

# You may uncomment the following lines if you want `ls' to be colorized:
# export LS_OPTIONS='--color=auto'
# eval "`dircolors`"
# alias ls='ls $LS_OPTIONS'
# alias ll='ls $LS_OPTIONS -l'
# alias l='ls $LS_OPTIONS -lA'
#
# Some more alias to avoid making mistakes:
# alias rm='rm -i'
# alias cp='cp -i'
# alias mv='mv -i'
alias ncmd='nagios_commander -n 127.0.0.1/nagios -u nagiosadmin -p fenestros'
```

Vérifiez l'utilisation de l'alias :

```
root@nagios:~# ncmd -q list -h
Hostname    Status
linux      UP
localhost  UP
```

Il est possible de spécifier un hôte à interroger en utilisant l'option **-h** :

```
root@nagios:~# ncmd -q list -h localhost
Fetching services and health on localhost
---
Service  State
---
ssh      OK
```

```
www         OK
```

Les autres options du script sont :

- **-H** - permet d'interroger un Groupe d'Hôtes,
- **-s** - permet d'interroger un service,
- **-S** - permet d'interroger un Groupe de Services,
- **-c** - permet de modifier la configuration de Nagios ou de spécifier des downtimes,
- **-q** - permet de préciser un type de requête.

Un exemple d'utilisation de l'option **-c** est la désactivation/activation des notifications :

```
root@nagios:~# ncmd -c set notifications disable
Your command request was successfully submitted to Nagios for processing.
notifications:disabled
root@nagios:~#
root@nagios:~# ncmd -c set notifications enable
Your command request was successfully submitted to Nagios for processing.
notifications:enabled
```

nagios-cli

Pour installer les dépendances necéssaires, lancez la commande suivante :

```
root@nagios:~# apt-get -y install patch python python-pip libpython-dev libncurses-dev libreadline-dev git
```

Utilisez **pip** pour installer le paquet Python **readline** :

```
root@nagios:~# pip install readline
```

Maintenant, téléchargez les paquets sources pour **nagios-cli** en utilisant la commande **git** :

```
root@nagios:~# git clone https://github.com/tehmaze/nagios-cli.git
```

```
Cloning into 'nagios-cli'...
remote: Counting objects: 390, done.
remote: Total 390 (delta 0), reused 0 (delta 0), pack-reused 390
Receiving objects: 100% (390/390), 73.34 KiB | 0 bytes/s, done.
Resolving deltas: 100% (236/236), done.
Checking connectivity... done.
```

Installez nagios-cli :

```
root@nagios:~# cd nagios-cli ; python setup.py install
```

Créez le fichier **/usr/local/nagios/etc/nagios-cli.cfg** :

```
root@nagios:~/nagios-cli# vi /usr/local/nagios/etc/nagios-cli.cfg
root@nagios:~/nagios-cli# cat /usr/local/nagios/etc/nagios-cli.cfg
[nagios]
log                     = /usr/local/nagios/var
command_file            = %(log)s/rw/nagios.cmd
log_file                = %(log)s/nagios.log
object_cache_file       = %(log)s/objects.cache
status_file             = %(log)s/status.dat
```

Lancez nagios-cli ainsi :

```
root@nagios:~/nagios-cli# nagios-cli -c /usr/local/nagios/etc/nagios-cli.cfg
Welcome to the nagios command line interface
nagios >
```

Pour connaître les commandes admises par nagios-cli, utilsez la commande **hgelp** :

```
nagios > help
Global commands:
..          EOF         about       configure  exit        help        host
license     quit        tail
```

```
Local commands:
list       ls
```

Pour lister les hôtes, utilisez la comande **list** ou simplement **ls** :

```
nagios > list
linux      localhost
nagios > ls
linux      localhost
```

Pour sélectionner le contexte d'un hôte particulier, utilisez la commande **host** :

```
nagios > host localhost
nagios (host) localhost>
```

Pour lister les services de l'hôte, utilisez la commande **ls** :

```
nagios (host) localhost> ls
ssh  www
```

Pour sélectionner le contexte d'un service spécifique, utilisez la commande **service** :

```
nagios (host) localhost> service ssh
nagios (host) localhost → ssh>
```

La commande **EOF** permet de retourner au contexte précédent :

```
nagios (host) localhost → ssh> EOF

nagios (host) localhost>
```

Etant dans le contexte de l'hôte, il est possible de se renseigner sur le statut de cet hôte en utilisant la commande **status** :

```
nagios (host) localhost> status
host name           : localhost
current state       : ↑ OK
plugin output       : PING OK - Packet loss = 0%, RTA = 0.14 ms
is flapping         : 0
last check          : 2018-05-15 15:01:15
last time down      : never
last state change   : 2018-04-24 15:07:16
check period        : 24x7
notification period : 24x7
current attempt     : 1
max attempts        : 5
service             : ssh                 ↑ OK
service             : www                 ↑ OK
```

Comprendre les Vérifications de Nagios

Nagios utilise des commandes externes pour vérifier si un hôte ou un service fonctionne correctement. C'est la responsabilité de la commande de procéder à la vérification puis de retourner un code retour à Nagios en fonction du résultat obtenu, agrémenté d'informations supplémentaires éventuelles. De ce fait, les codes retour des plugins Nagios sont standardisés :

Code Retour	**Statut**	**Description**
0	OK	Fonctionnenemnt normal.
1	WARNING	Un avertissement tel des ressources faibles.
2	CRITICAL	Ne fonctione pas correctement ou requiert une action.
3	UNKNOWN	Le plugin n'était pas en mesure de déterminer l'état de l'hôte ou du service.

Le format de la sortie de la commande n'est pas formaté par Nagios et prend la forme suivante :

```
Statut du PLUGIN - description du statut
```

Par exemple :

```
PING OK - Packet loss = 0%, RTA = 0.18 ms
DNS WARNING: 0.015 seconds response time
DISK CRITICAL - free space: /boot 18 MB (8% inode=99%)
```

De même, les options acceptées par les plugins standards de Nagios sont :

Option courte	**Option longue**	**Description**
-h,	–help	Fournir de l'aide en ligne.
-V,	–version	Imprimer à l'écran la version du plugin.
-v,	–verbose	Augmenter la verbosité de la sortie du plugin.
-t,	–timeout	Forcer le plugin à retourner l'état CRITICAL après le nombre de secondes spécifié.
-w,	–warning	Fournir les limites spécifiques au plugin pour le statut WARNING.
-c,	–critical	Fournir les limites spécifiques au plugin pour le statut CRITICAL.
-H,	–hostname	Fournir le nom d'hôte, l'adresse IP ou le socket Unix avec lequel il faut communiquer.
-4,	–use-ipv4	Utiliser IPv4.
-6,	–use-ipv6	Utiliser IPv6.

Tandis que la plupart des plugins liés au réseau acceptent les options suivantes :

Option courte	**Option longue**	**Description**
-p,	–port	Le port TCP ou UDP de la connexion.
-w,	–warning	Le temps en secondes après lequel un état de WARNING sera notifié.
-c,	–critical	Le temps en secondes après lequel un état de CRITICAL sera notifié.
-s,	–send	La chaîne qui sera envoyée au serveur.
-e,	–expect	La chaîne qui sera renvoyée par le serveur.
-q,	–quit	La chaîne qui sera envoyée au serveur pour rompre la connexion.
-A,	–all	Dans le cas où de multiples paramètres **–expect** sont attendus, cette option indique que **toutes** les réponses doivent être reçues. En l'abscence de cette option au moins une des réponses doit être reçue.
-m,	–maxbytes	Le nombre maximal d'octets à lire de la chaîne **–expect**.
-d,	–delay	Le délai en secondes entre **–send** et **–expect**.
-r,	–refuse	Le statut qui doit être indiqué en cas de refus de la connexion (**ok**, **warn** ou **crit**. Par défaut la valeur est **crit**).

Option courte	Option longue	Description
-M	S/O	Le statut qui doit être indiqué au cas où la réponse attendue n'est pas retournée par le serveur (**ok**, **warn** ou **crit**. Par défaut la valeur est **warn**).
-j,	–jail	Ne pas retourner la sortie du serveur dans la sortie du plugin.
-D,	–certificate	Le nombre de jours que le certificat SSL doit encore être valide. Cette option nécessite l'option **-ssl**.
-S,	–ssl	Etablir la connexion en utilisant SSL.
-E,	–escape	Ne pas retourner la sortie du serveur dans la sortie du plugin.

LAB #3 - Installation des Serveurs dans la VM CentOS 7

Préparation

A Faire : Arrêtez la VM CentOS 7. Dans l'interface de VirtualBox, configurez le réseau en NAT. Démarrez ensuite la VM CentOS 7.

Le Serveur Web Apache

Présentation d'Apache

Un serveur web est une machine doté d'un logiciel serveur qui attend des requêtes de la part de machines clientes afin de leur livrer de documents de types différents.

En 1994 le développement du serveur web le plus connue à l'époque, le démon **HTTP**, a été arrêté suite au départ de la NCSA de son principal développeur, **Rob McCool**.

Au début de l'année 1995, un groupe de webmestres indépendants s'est mis en place sous la direction de **Brian Behlendorf** et **Cliff Skolnick** pour reprendre le travail sur ce démon. Ce projet a pris le nom **Apache**. En même temps la NCSA a repris son propre travail de développement sur son démon HTTP. L'arrivée dans le groupe Apache de deux personnes de la NCSA en tant que membres honoraires, **Brandon Long** et **Beth Frank** a

permis la mise en commun des connaissances des deux groupes.

Le projet **Apache** est un projet de développement d'un serveur web libre pour les plateformes Unix et Windows™. La première version *officielle*, la 0.6.2 est sortie en avril 1995.

La **Fondation Apache**, créée en 1999 par l'équipe Apache, gère aujourd'hui non seulement le projet Apache mais aussi un grand nombre d'autres projets. La liste des projets de la Fondation peut être trouvée **ici**.

Apache est modulaire. Certains modules fondamentaux conditionnent comment Apache traite la question du multitraitement. Les modules multitraitements - **MPM - Multi-Processing Modules** - sont différents selon le système d'exploitation utilisé et la charge attendue.

- **mpm-winnt** - module propre à Windows™ qui utilise son support réseau natif,
- **prefork** - module propre à Unix et Linux qui implémente un serveur mono-tâche à duplication,
- **perchild** - module propre à Unix et Linux qui implémente un serveur autorisant des démons servant les requêtes à être assigner à plusieurs id utilisateurs,
- **worker** - module propre à Unix et Linux qui implémente un serveur hybride multi-tâche et multitraitement.

Ces modules sont compilés statiquement au binaire Apache et sont mutuellement exclusifs.

Installation

Cette installation est à effectuer dans votre client CentOS 7.

Sous **CentOS 7**, Apache n'est pas installé par défaut :

```
[root@centos7 ~]# rpm -qa | grep httpd
[root@centos7 ~]#
[root@centos7 ~]# yum install httpd
```

Le service n'est pas configuré pour démarrer automatiquement :

```
[root@centos7 ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
```

```
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
     Docs: man:httpd(8)
           man:apachectl(8)
```

Saisissez donc les commandes suivantes et vérifiez le résultat :

```
[root@centos7 ~]# systemctl enable httpd.service
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to
/usr/lib/systemd/system/httpd.service.
[root@centos7 ~]# systemctl start httpd.service
[root@centos7 ~]# systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2017-04-27 13:09:01 CEST; 1s ago
     Docs: man:httpd(8)
           man:apachectl(8)
 Main PID: 9861 (httpd)
   Status: "Processing requests..."
   CGroup: /system.slice/httpd.service
           ├─9861 /usr/sbin/httpd -DFOREGROUND
           ├─9862 /usr/sbin/httpd -DFOREGROUND
           ├─9863 /usr/sbin/httpd -DFOREGROUND
           ├─9864 /usr/sbin/httpd -DFOREGROUND
           ├─9865 /usr/sbin/httpd -DFOREGROUND
           └─9866 /usr/sbin/httpd -DFOREGROUND

Apr 27 13:09:01 centos7.fenestros.loc systemd[1]: Starting The Apache HTTP Se...
Apr 27 13:09:01 centos7.fenestros.loc systemd[1]: Started The Apache HTTP Ser...
Hint: Some lines were ellipsized, use -l to show in full.
```

Testez le serveur apache avec telnet

Telnet n'est pas installé par défaut sous CentOS 7 :

```
[root@centos7 ~]# which telnet
/usr/bin/which: no telnet in
(/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin:/usr/lib/jvm/jre-1.8.0-
openjdk-1.8.0.121-0.b13.el7_3.x86_64/bin:/usr/lib/jvm/jre-1.8.0-
openjdk-1.8.0.131-2.b11.el7_3.x86_64/bin:/usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.131-2.b11.el7_3.x86_64/bin)
```

Installez donc telnet :

```
[root@centos7 ~]# yum install telnet
```

Testez maintenant le service Apache :

```
[root@centos7 ~]# telnet localhost 80
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
```

Tapez ensuite **GET /** :

```
GET /
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd"><html><head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
        <title>Apache HTTP Server Test Page powered by CentOS</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

    <!-- Bootstrap -->
    <link href="/noindex/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="noindex/css/open-sans.css" type="text/css" />
```

```
<style type="text/css"><!--

body {
  font-family: "Open Sans", Helvetica, sans-serif;
  font-weight: 100;
  color: #ccc;
  background: rgba(10, 24, 55, 1);
  font-size: 16px;
}

h2, h3, h4 {
...
```

Pour voir la version d'Apache installée, utilisez la comande suivante :

```
[root@centos7 ~]# httpd -v
Server version: Apache/2.4.6 (CentOS)
Server built:   Apr 12 2017 21:03:28
```

Dernièrement, créez une page index.html :

```
[root@linux ~]# vi /var/www/html/index.html
[root@linux ~]# cat /var/www/html/index.html
<html>
<title>Test page for Nagios</title>
<body>
Just a test page!
</body>
</html>
```

Le Serveur de Bases de Données MariaDB

Présentation

MariaDB est un fork de **MySQL** et est inclut dans les dépôts de **CentOS 7**.

Installation

Pour installer MariaDB, utilisez yum :

```
[root@centos7 ~]# yum install mariadb mariadb-server
```

Ensuite démarrez votre serveur MariaDB :

```
[root@centos7 ~]# systemctl status mariadb.service
● mariadb.service - MariaDB database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
[root@centos7 ~]# systemctl enable mariadb.service
Created symlink from /etc/systemd/system/multi-user.target.wants/mariadb.service to
/usr/lib/systemd/system/mariadb.service.
[root@centos7 ~]# systemctl start mariadb.service
[root@centos7 ~]# systemctl status mariadb.service
● mariadb.service - MariaDB database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2017-04-27 13:44:02 CEST; 3s ago
  Process: 30380 ExecStartPost=/usr/libexec/mariadb-wait-ready $MAINPID (code=exited, status=0/SUCCESS)
  Process: 30295 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exited, status=0/SUCCESS)
 Main PID: 30379 (mysqld_safe)
   CGroup: /system.slice/mariadb.service
           ├─30379 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
           └─30536 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql --plugin-
dir=/usr/lib64/mysql/plugin --log-error=/var/log/mariadb/mariadb.l...
```

```
Apr 27 13:44:00 centos7.fenestros.loc mariadb-prepare-db-dir[30295]: OK
Apr 27 13:44:00 centos7.fenestros.loc mariadb-prepare-db-dir[30295]: To start mysqld at boot time you have to
copy
Apr 27 13:44:00 centos7.fenestros.loc mariadb-prepare-db-dir[30295]: support-files/mysql.server to the right
place for your system
Apr 27 13:44:00 centos7.fenestros.loc mariadb-prepare-db-dir[30295]: PLEASE REMEMBER TO SET A PASSWORD FOR THE
MariaDB root USER !
Apr 27 13:44:00 centos7.fenestros.loc mariadb-prepare-db-dir[30295]: To do so, start the server, then issue the
following commands:
Apr 27 13:44:00 centos7.fenestros.loc mariadb-prepare-db-dir[30295]: '/usr/bin/mysqladmin' -u root password 'new-
password'
Apr 27 13:44:00 centos7.fenestros.loc mariadb-prepare-db-dir[30295]: '/usr/bin/mysqladmin' -u root -h
centos7.fenestros.loc password 'new-password'
Apr 27 13:44:00 centos7.fenestros.loc mysqld_safe[30379]: 170427 13:44:00 mysqld_safe Logging to
'/var/log/mariadb/mariadb.log'.
Apr 27 13:44:00 centos7.fenestros.loc mysqld_safe[30379]: 170427 13:44:00 mysqld_safe Starting mysqld daemon with
databases from /var/lib/mysql
Apr 27 13:44:02 centos7.fenestros.loc systemd[1]: Started MariaDB database server.
```

Configuration

Lors de l'installation MariaDB n'a pas de mot de passe attibué à l'utilisateur root. Créez donc le mot de passe **fenestros** puis téstez la connection à MariaDB :

```
[root@centos7 ~]# mysqladmin -u root password fenestros
[root@centos7 ~]# mysql
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)
[root@centos7 ~]# mysql -uroot -p
Enter password: fenestros
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 5.5.52-MariaDB MariaDB Server
```

```
Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> exit
Bye
```

Afin de permettre à Nagios de vérifier le bon fonctionnement de MariaDB, créez une base de données dénommée **nagios** :

```
[root@linux ~]# mysql -uroot -p
Enter password: fenestros
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 460
Server version: 5.5.56-MariaDB MariaDB Server

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| test               |
+--------------------+
4 rows in set (0.01 sec)

MariaDB [(none)]> CREATE DATABASE nagios;
Query OK, 1 row affected (0.28 sec)

MariaDB [(none)]> SHOW DATABASES;
```

```
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| nagios             |
| performance_schema |
| test               |
+--------------------+
5 rows in set (0.01 sec)

MariaDB [(none)]>
```

Créez ensuite une utilisateur dénommé **nagios** ayant toutes les privilèges sur la base de données **nagios** et un mot de passe **password** :

```
MariaDB [(none)]> GRANT ALL ON nagios.* to 'nagios'@'%' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> exit
```

Le Serveur SMTP Postfix

Présentation

La messagerie utilise les protocols suivants :

- SMTP (Simple Message Transfer Protocol),
- POP (Post Office Protocol),
- IMAP (Internet Message Access Protocol).

Lors de l'utilisation du protocol, c'est l'expéditeur qui initie le transfert tandis qu'avec les protocols POP et IMAP c'est le destinataire qui initie la collecte.

Un serveur SMTP est aussi appelé un MTA (Mail Transfer Agent) tandis que les serveurs POP et IMAP sont souvents appelés des *Pull Mail Servers*. Enfin les clients de messagerie sont des MUA (Mail User Agent).

Dans un système Linux, le mail est stocké pour chaque utilisateur soit dans le répertoire **/var/spool/mail**, soit dans un répertoire dans le répertoire personnel de chaque utilisateur.

Les quatre MTA les plus utilisés sous Linux sont :

- **Sendmail**,
- **Postfix**,
- **Exim**,
- **Qmail**.

Postfix est considéré d'être un des MTA le plus facilement configuré par 250 directives. Ses fichiers sont facilement lisibles par l'être humain ce qui n'est pas le cas de sendmail.

Les deux Pull Mail Servers les plus utilisés sous Linux sont :

- **Cyrus IMAP**,
- **Dovecot**,
- **Fetchmail**.

Fetchmail remplit un rôle spécifique et n'est utilisé que quand le serveur n'est pas connecté en permanance à Internet.

Les quatre MUA les plus utilisés sous Linux sont :

- **http://projects.gnome.org/evolution/Evolution**,
- **KMail**,
- **Thnderbird**,
- **mutt**.

Deux utilitaires simples permettent la lecture des spools de mail locaux en ligne de commande aussi bien que l'envoie des messages :

- **mail**,
- **nail**.

La commande **nail** diffère de la commande **mail** par le fait qu'elle peut gérer des fichiers attachés.

La commande mail est souvent un lien symbolique ver la commande **mailx** :

```
[root@linux ~]# ls -l /bin/mail
lrwxrwxrwx. 1 root root 5 Apr 23 14:10 /bin/mail -> mailx
```

Les options de la commande mailx sont :

```
[root@linux ~]# mailx --help
mailx: illegal option -- -
Usage: mailx -eiIUdEFntBDNHRVv~ -T FILE -u USER -h hops -r address -s SUBJECT -a FILE -q FILE -f FILE -A ACCOUNT
-b USERS -c USERS -S OPTION users
```

La syntaxe de la commande mailx dans le cas d'un envoie est :

```
mailx [-s  objet ] [-c  ccadresse ] [-b  bccadresse ] adresse_destinataire
```

Lors de la lecture du spool de mail local, la syntaxe est la suivante :

```
mailx [-f [ spool de mail local ] | -u  nom_utilisateur ]
```

Par exemple :

```
[root@linux ~]# mail -s "test  message" -c trainee root
This is a test message
[^D]
EOT
```

```
[root@linux ~]# su - trainee
[trainee@linux ~]$ mail
Heirloom Mail version 12.5 7/5/10.  Type ? for help.
"/var/spool/mail/trainee": 1 message 1 new
>N  1 root                  Sun May 20 09:21  19/623   "test  message"
& 1
Message  1:
From root@linux.i2tch.loc  Sun May 20 09:21:23 2018
Return-Path: <root@linux.i2tch.loc>
X-Original-To: trainee
Delivered-To: trainee@linux.i2tch.loc
Date: Sun, 20 May 2018 09:21:23 +0200
To: root@linux.i2tch.loc
Subject: test  message
Cc: trainee@linux.i2tch.loc
User-Agent: Heirloom mailx 12.5 7/5/10
Content-Type: text/plain; charset=us-ascii
From: root@linux.i2tch.loc (root)
Status: R

This is a test message

& q
Held 1 message in /var/spool/mail/trainee
```

Il est aussi possible d'injecter le contenu d'un fichier sur l'entrée standard de la commande mailx afin de l'utiliser comme le contenu du message :

```
[trainee@linux ~]$ echo i2tch > mail.txt
You have mail in /var/spool/mail/trainee
[trainee@linux ~]$ mail -s "test message back" < mail.txt root
[trainee@linux ~]$ exit
logout
You have new mail in /var/spool/mail/root
[root@linux ~]# mail
```

```
Heirloom Mail version 12.5 7/5/10.  Type ? for help.
"/var/spool/mail/root": 4 messages 4 new
>N  1 trainee@centos7.fene  Sat Apr 30 12:38  16/688   "*** SECURITY informat"
 N  2 trainee@centos7.fene  Mon Apr 23 12:04  16/688   "*** SECURITY informat"
 N  3 root                  Sun May 20 09:21  19/617   "test  message"
 N  4 trainee               Sun May 20 09:23  18/591   "test message back"
& 4
Message  4:
From trainee@linux.i2tch.loc  Sun May 20 09:23:40 2018
Return-Path: <trainee@linux.i2tch.loc>
X-Original-To: root
Delivered-To: root@linux.i2tch.loc
Date: Sun, 20 May 2018 09:23:40 +0200
To: root@linux.i2tch.loc
Subject: test message back
User-Agent: Heirloom mailx 12.5 7/5/10
Content-Type: text/plain; charset=us-ascii
From: trainee@linux.i2tch.loc (trainee)
Status: R

i2tch

& q
Held 4 messages in /var/spool/mail/root
```

Configuration de votre Machine Virtuelle

Comme vous allez utiliser le nom de domaine **mail.i2tch.com** pour votre serveur postfix, modifiez votre fichier **/etc/hosts** ainsi :

```
root@nagios:~# vi /etc/hosts
root@nagios:~# vi /etc/hosts
root@nagios:~# cat /etc/hosts
127.0.0.1   localhost
```

```
127.0.1.1   debian8.fenestros.loc    debian8
10.0.2.14   mail.i2tch.com

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Installation

Installez **postfix** et **procmail** :

```
[root@linux ~]# yum install postfix procmail
```

Configuration de Base

Le fichier /etc/postfix/main.cf

Utilisez les commandes suivantes pour voir les directives actives dans le fichiers **/etc/postfix/main.cf** :

```
[root@linux ~]# cd /tmp ; grep -E -v '^(#|$)'  /etc/postfix/main.cf > main.cf
You have mail in /var/spool/mail/root
[root@linux tmp]# cat main.cf
queue_directory = /var/spool/postfix
command_directory = /usr/sbin
daemon_directory = /usr/libexec/postfix
data_directory = /var/lib/postfix
mail_owner = postfix
inet_interfaces = localhost
inet_protocols = all
```

```
mydestination = $myhostname, localhost.$mydomain, localhost
unknown_local_recipient_reject_code = 550
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases

debug_peer_level = 2
debugger_command =
     PATH=/bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin
     ddd $daemon_directory/$process_name $process_id & sleep 5
sendmail_path = /usr/sbin/sendmail.postfix
newaliases_path = /usr/bin/newaliases.postfix
mailq_path = /usr/bin/mailq.postfix
setgid_group = postdrop
html_directory = no
manpage_directory = /usr/share/man
sample_directory = /usr/share/doc/postfix-2.10.1/samples
readme_directory = /usr/share/doc/postfix-2.10.1/README_FILES
```

Ce fichier comporte des directives au formats suivants :

- paramètre = valeur
- autre_paramètre = $paramètre

Copiez votre fichier main.cf en main.old

```
[root@linux tmp]# cd ~
[root@linux ~]# cp /etc/postfix/main.cf /etc/postfix/main.old
```

Modifiez **/etc/postfix/main.cf** ainsi :

```
[root@linux ~]# vi /etc/postfix/main.cf
[root@linux ~]# cat /etc/postfix/main.cf
myhostname = mail.i2tch.com
mydomain= i2tch.com
```

```
myorigin = $mydomain
mynetworks = 10.0.2.0/24, 127.0.0.0/8
mail_spool_directory = /var/spool/mail
mailbox_command = /usr/bin/procmail -Y -a $DOMAIN
smtpd_banner = $myhostname ESMTP $mail_name ($mail_version)
delay_warning_time = 4h
recipient_delimiter = +
owner_request_special = no
mail_owner = postfix
inet_interfaces = all
mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
#unknown_local_recipient_reject_code = 550
unknown_local_recipient_reject_code = 450
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
debugger_command =
         PATH=/bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin
         xxgdb $daemon_directory/$process_name $process_id & sleep 5
sendmail_path = /usr/sbin/sendmail.postfix
newaliases_path = /usr/bin/newaliases.postfix
mailq_path = /usr/bin/mailq.postfix
setgid_group = postdrop
manpage_directory = /usr/share/man
sample_directory = /usr/share/doc/postfix-2.10.1/samples
readme_directory = /usr/share/doc/postfix-2.10.1/README_FILES
queue_directory = /var/spool/postfix
command_directory = /usr/sbin
daemon_directory = /usr/libexec/postfix
```

Les directives dans l'exemple ci-dessus sont :

Directive	**Description**
myhostname	Le nom de machine Internet de ce système de messagerie.
mydomain	Le nom de domaine Internet du système de messagerie.

Directive	Description
myorigin	Le domaine par défaut utilisé pour les messages postés localement.
mynetworks	La liste des clients SMTP “internes” qui ont plus de privilèges que les “étrangers”.
mail_spool_directory	Le répertoire où les boîtes-aux-lettres locales sont stockées.
mailbox_command	Commande externe optionnelle que l'agent de livraison local doit utiliser pour la livraison des messages.
smtpd_banner	Texte qui suit le code de statut 220 dans la bannière d'accueil.
delay_warning_time	Temps au delà duquel l'expéditeur reçoit les en-têtes d'un message toujours en file d'attente.
recipient_delimiter	Le délimiteur système de l'extension de adresse de destination.
owner_request_special	Applique un traitement particulier aux parties locales des adresses de listes de propriétaires ou de requêtes.
relayhost	La machine par défaut où livrer le courrier extérieur.
mail_owner	Le compte du système qui possède la file d'attente et la plupart des processus démons de Postfix.
inet_interfaces	Les adresses réseau par lesquelles le système de messagerie reçoit les messages.
mydestination	Liste des domaines livrés par le transporteur de messages.
unknown_local_recipient_reject_code	Code numérique de réponse du serveur SMTP de Postfix lorsque le destinataire n'est pas trouvé.
alias_maps	La base de données des alias utilisée pour la livraison locale.
alias_database	La base de données des alias pour la livraison locale.
debugger_command	La commande externe à exécuter lorsqu'un programme démon de Postfix est invoké avec l'option -D.
sendmail_path	Indique l'emplacement de la commande sendmail de Postfix.
newaliases_path	Indique l'emplacement de la commande newaliases.
mailq_path	Indique où est installée la commande Postfix mailq. Cette commande peut être utilisée pour afficher la file d'attente.
setgid_group	Le groupe propriétaire des commandes set-gid de Postfix et des répertoires en écriture pour le groupe.
manpage_directory	Emplacement des pages de manuel de Postfix.
sample_directory	Emplacement des exemples de fichiers de configuration de Postfix.
readme_directory	Emplacement des fichiers README de Postfix.
queue_directory	Emplacement du répertoire racine de la file d'attente de Postfix.
command_directory	Emplacement des commandes administratives de Postfix.
daemon_directory	Emplacement des démons Postfix.

A faire - Pour plus d'informations concernant les directives, consultez **cette page**.

Les options les plus importantes de la commande sendmail de postfix sont :

```
       -Am (ignored)

       -Ac (ignored)
              Postfix sendmail uses the same configuration file regardless  of
              whether or not a message is an initial submission.

       -B body_type
              The message body MIME type: 7BIT or 8BITMIME.

       -bd    Go  into  daemon  mode. This mode of operation is implemented by
              executing the "postfix start" command.

       -bh (ignored)

       -bH (ignored)
              Postfix has no persistent host status database.

       -bi    Initialize alias database. See the newaliases command above.

       -bm    Read mail from standard input and arrange for delivery.  This is
              the default mode of operation.

       -bp    List the mail queue. See the mailq command above.

       -bs    Stand-alone  SMTP  server mode. Read SMTP commands from standard
              input, and write responses to standard output.   In  stand-alone
              SMTP  server  mode,  mail relaying and other access controls are
              disabled by default. To enable them,  run  the  process  as  the
              mail_owner user.

              This  mode  of  operation is implemented by running the smtpd(8)
              daemon.
```

```
       -bv    Do not collect or deliver a  message.  Instead,  send  an  email
              report  after  verifying each recipient address.  This is useful
              for testing address rewriting and routing configurations.

              This feature is available in Postfix version 2.1 and later.
```

Testez votre fichier de configuration avec la commande **postfix** :

```
[root@linux ~]# postfix check
[root@linux ~]#
```

Important - Consultez le manuel de la commande postfix pour connaître ses options et ses arguments.

Consultez l'arborescence du répertoire **/var/spool/postfix** :

```
[root@linux ~]# ls /var/spool/postfix
active  bounce  corrupt  defer  deferred  flush  hold  incoming  maildrop  pid  private  public  saved  trace
```

Comprendre la fonction des répertoires dans **/var/spool/postfix** nécessite une compréhension des processus principaux de postfix :

Processus	**Description**
master	Processus central de postfix. Il lance les autres processus. Il est lancé par root. Ses paramètres de configuration se trouvent dans le fichier **/etc/postfix/master.cf**
qmgr	Processus qui lit la queue **incoming** et place une partie des messages dans **active**. Ensuite il efface les messages où le traitement s'est bien passé. Dans le cas contraire, il place les messages dans **deferred**.
pickup	Processus qui attend d'être informé par **postdrop** de la présence de nouveaux messages dans le répertoire **maildrop**. Il passe ensuite les messages au processus **cleanup**.
smtpd	Processus qui reçoit les messages de l'extérieur et les passe au processus **cleanup**.

Processus	**Description**
cleanup	Processus qui reçoit les messages de **pickup** ou de **smtpd** et qui les complète en termes de champs manquants (p.e. From: To: etc) tout en éliminant les doublons d'adresses destinataires. Il délègue au processus **trivial-rewrite** la tâche de transformer les adresses de l'enveloppe et des en-têtes d'adresses de type nom@fqdn.extension. Il place les messages traités dans le répertoire **incoming** et informe le processus **qmgr** qu'il faut examiner ce dernier.
bounce	Processus qui délivre des messages de notification en cas d'échec définitif, de remise différée, de remise avec succès ou de vérifications d'adresses. IL maintient dans le répertoire **bounce** des informations sur les raisons des rejets des messages.
defer	Un alias du processus bounce qui maintient dans le répertoire **defer** les informations d'explications des raisons des messages différés.
trace	Un alias du processus bounce qui maintient dans le répertoire **trace** les informations de suivi de la remise des messages si ces informations ont été demandées en utilisant la commande **sendmail -bv** ou **sendmail -v**.
flush	Processus qui constitue une liste de messages, corréspondants à la directive du fichier **/etc/postfix/main.cf fast_flush_domain**, qui vont être traités plus prioritairement.
trivial-rewrite	Voir les processus **cleanup** et **qmgr**.
verify	Processus qui maintient une base d'addresses connues valides ou invalides.
scache	Processus qui maintient un cache des serveurs extérieurs où le processus smtpd a pu se connecter. Ces informations sont gardés pendant le temps spécifié par la directive **max_idle**.
anvil	Processus qui est chargé de la collecte des statistiques du nombre de connexions et de requêtes éffectuées par chaque client.
showcase	Processus qui rapporte l'état des files d'attente.

Après avoir vu les processus de postfix, nous pouvons se concentrer sur les répertoires présents dans **/var/spool/postfix** :

Répertoire	**Contenu**
active	Répertoire de file d'attente. Contient les messages en cours de traitement. En réalité ce répertoire est vide car les messages concernés sont tous en mémoire.
bounce	Répertoire contenant les raisons des rejets des messages.
corrupt	Répertoire contenant des messages corrompus.
defer	Répertoire de stockage temporaire. Contient les informations sur la raison des échecs des messages qui se trouvent dans le répertoire **deferred**.
deferred	Répertoire de file d'attente. Contient des sous-répertoires qui contiennent les messages qui n'ont pas pu être remis. Chaque sous-répertoire est nommé après le premier caractère de la **Queue ID** du message.
flush	Répertoire utilisé par le processus **flush**.
hold	Répertoire de stockage temporaire. Voir ci-dessus.
incoming	Répertoire de file d'attente. Contient les messages placés par le processus **cleanup**.

Répertoire	**Contenu**
maildrop	Répertoire de stockage temporaire. Contient des messages crées localement.
pid	Répertoire de stockage temporaire. Contient les PID des processus postfix lancés.
private	Répertoire contenant une liste de sockets disponibles pour des utilisateurs priviligiés.
public	Répertoire contenant une liste de sockets disponibles pour tout le monde.
trace	Répertoire utilisé par le processus **trace**.

Modifiez maintenant les droits sur le répertoire **/var/spool/mail**:

```
[root@linux ~]# chmod 1777 /var/spool/mail
```

Re-démarrez le serveur postfix :

```
[root@linux ~]# systemctl restart postfix
[root@linux ~]# systemctl status postfix
● postfix.service - Postfix Mail Transport Agent
   Loaded: loaded (/usr/lib/systemd/system/postfix.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2018-05-20 09:34:53 CEST; 6s ago
  Process: 11558 ExecStop=/usr/sbin/postfix stop (code=exited, status=0/SUCCESS)
  Process: 11573 ExecStart=/usr/sbin/postfix start (code=exited, status=0/SUCCESS)
  Process: 11570 ExecStartPre=/usr/libexec/postfix/chroot-update (code=exited, status=0/SUCCESS)
  Process: 11567 ExecStartPre=/usr/libexec/postfix/aliasesdb (code=exited, status=0/SUCCESS)
 Main PID: 11657 (master)
   CGroup: /system.slice/postfix.service
           ├─11657 /usr/libexec/postfix/master -w
           ├─11659 pickup -l -t unix -u
           └─11660 qmgr -l -t unix -u

May 20 09:34:51 linux.i2tch.loc systemd[1]: Starting Postfix Mail Transport ....
May 20 09:34:53 linux.i2tch.loc postfix/master[11657]: daemon started -- vers...
May 20 09:34:53 linux.i2tch.loc systemd[1]: Started Postfix Mail Transport A....
Hint: Some lines were ellipsized, use -l to show in full.
```

Testez maintenant le serveur smtp de postfix en envoyant un message de root à trainee :

```
[root@linux ~]# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 mail.i2tch.com ESMTP Postfix (2.10.1)
HELO me
250 mail.i2tch.com
MAIL from: root@i2tch.com
250 2.1.0 Ok
RCPT to: trainee@i2tch.com
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Subject: Test email
This is a test
.
250 2.0.0 Ok: queued as 6DDAF3344BA6
QUIT
221 2.0.0 Bye
Connection closed by foreign host.
[root@linux ~]#
```

Notez :

- le code **220** qui indique sue le serveur attend des instructions,
- le déroulement de la conversation :
  - **HELO me** sert à vous identifier,
  - **MAIL from: root@i2tch.com** indique l'expéditeur,
  - **RCPT to: trainee@i2tch.com** indique le destinataire,
  - **DATA** indique que ce qui suit est le message,
- le code **250** qui indique que la commande s'est bien déroulée.
- le point sur une ligne vide indique la fin de la section DATA.

Consultez maintenant le fichier **/var/log/maillog**. Vous devez constater que votre message a été livré à trainee :

```
[root@linux ~]# tail /var/log/maillog
May 20 09:42:31 linux postfix/smtpd[15018]: connect from localhost.localdomain[127.0.0.1]
May 20 09:43:10 linux postfix/smtpd[15018]: 49EDD3344BA6: client=localhost.localdomain[127.0.0.1]
May 20 09:43:34 linux postfix/cleanup[15318]: 49EDD3344BA6: message-
id=<20180520074310.49EDD3344BA6@mail.i2tch.com>
May 20 09:43:34 linux postfix/qmgr[11660]: 49EDD3344BA6: from=<root@i2tch.com>, size=342, nrcpt=1 (queue active)
May 20 09:43:34 linux postfix/local[15468]: 49EDD3344BA6: to=<trainee@i2tch.com>, relay=local, delay=38,
delays=38/0.07/0/0.05, dsn=2.0.0, status=sent (delivered to command: /usr/bin/procmail -Y -a $DOMAIN)
May 20 09:43:34 linux postfix/qmgr[11660]: 49EDD3344BA6: removed
May 20 09:43:38 linux postfix/smtpd[15018]: disconnect from localhost.localdomain[127.0.0.1]
```

Important : Il est possible a tout moment de visualiser le contenu du spool de mail en utilisant la commande **mailq** qui est équivalente à la commande **sendmail bp**.

Définir les Aliases

Les deux lignes suivantes, issues du fichier **/etc/postfix/main.cf** indiquent l'emplacement des fichiers relatifs aux **aliases** :

```
...
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
...
```

Voici le contenu du fichier **/etc/aliases** :

```
[root@linux ~]# cat /etc/aliases
#
```

```
#  Aliases in this file will NOT be expanded in the header from
#  Mail, but WILL be visible over networks or from /bin/mail.
#
#   >>>>>>>>>>    The program "newaliases" must be run after
#   >> NOTE >>    this file is updated for any changes to
#   >>>>>>>>>>    show through to sendmail.
#

# Basic system aliases -- these MUST be present.
mailer-daemon:  postmaster
postmaster: root

# General redirections for pseudo accounts.
bin:        root
daemon:     root
adm:        root
lp:     root
sync:       root
shutdown:   root
halt:       root
mail:       root
news:       root
uucp:       root
operator:   root
games:      root
gopher:     root
ftp:        root
nobody:     root
radiusd:    root
nut:        root
dbus:       root
vcsa:       root
canna:      root
wnn:        root
```

```
rpm:        root
nscd:       root
pcap:       root
apache:     root
webalizer:  root
dovecot:    root
fax:        root
quagga:     root
radvd:      root
pvm:        root
amandabackup:       root
privoxy:    root
ident:      root
named:      root
xfs:        root
gdm:        root
mailnull:   root
postgres:   root
sshd:       root
smmsp:      root
postfix:    root
netdump:    root
ldap:       root
squid:      root
ntp:        root
mysql:      root
desktop:    root
rpcuser:    root
rpc:        root
nfsnobody:  root

ingres:     root
system:     root
toor:       root
```

```
manager:    root
dumper:     root
abuse:      root

newsadm:    news
newsadmin:  news
usenet:     news
ftpadm:     ftp
ftpadmin:   ftp
ftp-adm:    ftp
ftp-admin:  ftp
www:        webmaster
webmaster:  root
noc:        root
security:   root
hostmaster: root
info:       postmaster
marketing:  postmaster
sales:      postmaster
support:    postmaster


# trap decode to catch security attacks
decode:     root

# Person who should get root's mail
#root:      marc
```

Il est important de spécifier un utilisateur existant qui recevra le mail de root et ceci pour des raisons légales liées à la boîte de mail **postmaster**, l'administrateur du serveur vu de l'extérieur.

Il est aussi possible de créer des aliases pour harmoniser les adresses email de l'organisation. Si, par exemple, l'adresse email doit être au format **prénom.nom** mais que les noms de comptes du système linux sont au format **prénom**, il convient de rajouter au fichier une ligne pour chaque personne au format suivant :

```
...
prénom.nom:     prénom
...
```

Modifiez donc votre fichier ainsi :

```
[root@linux ~]# vi /etc/aliases
[root@linux ~]# tail /etc/aliases


# trap decode to catch security attacks
decode:     root

# Person who should get root's mail
root:       trainee

# Employee Accounts
mickey.mouse:   trainee
```

Afin de prendre en compte la nouvelle liste d'alias, il faut utiliser la commande **newaliases** :

```
[root@linux ~]# newaliases
```

et demander à postfix de relire ses fichiers de configuration :

```
[root@linux ~]# systemctl reload postfix
```

En utilisant telnet, envoyez un message de **trainee** à **mickey.mouse**. Ce message arrivera dans la bôite de réception de trainee grâce à l'alias créé ci dessus :

```
[root@linux ~]# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
```

```
Escape character is '^]'.
220 mail.i2tch.com ESMTP Postfix (2.10.1)
HELO me
250 mail.i2tch.com
MAIL from: trainee@i2tch.com
250 2.1.0 Ok
RCPT to: mickey.mouse@i2tch.com
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Subject: Alias
This is a message sent to an alias
.
250 2.0.0 Ok: queued as 2D8553344BA7
QUIT
221 2.0.0 Bye
Connection closed by foreign host.
```

Consultez maintenant le journal **/var/log/maillog**. Vous apercevrez des lignes similaires à :

```
[root@linux ~]# tail /var/log/maillog
May 20 10:02:56 linux postfix/smtpd[23443]: disconnect from localhost.localdomain[127.0.0.1]
May 20 10:09:10 linux postfix/postfix-script[26547]: refreshing the Postfix mail system
May 20 10:09:10 linux postfix/master[11657]: reload -- version 2.10.1, configuration /etc/postfix
May 20 10:09:42 linux postfix/smtpd[26779]: connect from localhost.localdomain[127.0.0.1]
May 20 10:10:17 linux postfix/smtpd[26779]: 2D8553344BA7: client=localhost.localdomain[127.0.0.1]
May 20 10:10:43 linux postfix/cleanup[27039]: 2D8553344BA7: message-
id=<20180520081017.2D8553344BA7@mail.i2tch.com>
May 20 10:10:43 linux postfix/qmgr[26553]: 2D8553344BA7: from=<trainee@i2tch.com>, size=365, nrcpt=1 (queue
active)
May 20 10:10:43 linux postfix/local[27230]: 2D8553344BA7: to=<trainee@i2tch.com>,
orig_to=<mickey.mouse@i2tch.com>, relay=local, delay=45, delays=45/0.06/0/0.05, dsn=2.0.0, status=sent (delivered
to command: /usr/bin/procmail -Y -a $DOMAIN)
May 20 10:10:43 linux postfix/qmgr[26553]: 2D8553344BA7: removed
```

```
May 20 10:10:46 linux postfix/smtpd[26779]: disconnect from localhost.localdomain[127.0.0.1]
```

Notez la présence de la ligne suivante :

May 20 10:10:43 linux postfix/local[27230]: 2D8553344BA7: to=trainee@i2tch.com, orig_to=mickey.mouse@i2tch.com, relay=local, delay=45, delays=45/0.06/0/0.05, dsn=2.0.0, status=sent (delivered to command: /usr/bin/procmail -Y -a $DOMAIN)

Cette ligne démontre que l'alias fonctionne.

Important : Un utilisateur peut transférer son email vers un autre utilisateur du système ou bien vers une adresse email valide en inscrivant le nom ou l'adresse dans le fichier **~.forward**.

La Commande postconf

La commande **postconf** peut vous être très utile. Grâce à l'option **-d** vous pouvez visualiser les valeurs par défaut des directives de configuration de postfix au lieu des valeurs utilisées. Grâce à l'option **-n** vous pouvez visualiser les valeurs des directives de configuration de postfix qui sont différentes de valeurs par défaut :

```
[root@linux ~]# postconf -n
alias_database = hash:/etc/aliases
alias_maps = hash:/etc/aliases
command_directory = /usr/sbin
config_directory = /etc/postfix
daemon_directory = /usr/libexec/postfix
debugger_command = PATH=/bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin xxgdb $daemon_directory/$process_name
$process_id & sleep 5
delay_warning_time = 4h
inet_interfaces = all
mail_owner = postfix
mail_spool_directory = /var/spool/mail
```

```
mailbox_command = /usr/bin/procmail -Y -a $DOMAIN
mailq_path = /usr/bin/mailq.postfix
manpage_directory = /usr/share/man
mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
mydomain = i2tch.com
myhostname = mail.i2tch.com
mynetworks = 10.0.2.0/24, 127.0.0.0/8
myorigin = $mydomain
newaliases_path = /usr/bin/newaliases.postfix
owner_request_special = no
queue_directory = /var/spool/postfix
readme_directory = /usr/share/doc/postfix-2.10.1/README_FILES
recipient_delimiter = +
sample_directory = /usr/share/doc/postfix-2.10.1/samples
sendmail_path = /usr/sbin/sendmail.postfix
setgid_group = postdrop
smtpd_banner = $myhostname ESMTP $mail_name ($mail_version)
unknown_local_recipient_reject_code = 450
```

Le Serveur FTP vsftpd

Installation

Le paquet **vsftpd** *Very Secure FTP daemon* se trouve dans les dépôts CentOS.

```
[root@linux ~]# yum install vsftpd
```

Configurez le service **vsftpd** :

```
[root@linux ~]# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; vendor preset: disabled)
```

```
   Active: inactive (dead)
[root@linux ~]# systemctl enable vsftpd
Created symlink from /etc/systemd/system/multi-user.target.wants/vsftpd.service to
/usr/lib/systemd/system/vsftpd.service.
[root@linux ~]# systemctl start vsftpd
[root@linux ~]# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2018-05-20 11:07:07 CEST; 6s ago
  Process: 5399 ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf (code=exited, status=0/SUCCESS)
 Main PID: 5404 (vsftpd)
   CGroup: /system.slice/vsftpd.service
           └─5404 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

May 20 11:07:07 linux.i2tch.loc systemd[1]: Starting Vsftpd ftp daemon...
May 20 11:07:07 linux.i2tch.loc systemd[1]: Started Vsftpd ftp daemon.
```

Configuration de base

Le fichier de configuration de vsftpd est **/etc/vsftpd/vsftpd.conf** :

```
[root@linux ~]# cat /etc/vsftpd/vsftpd.conf
# Example config file /etc/vsftpd/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
```

```
anonymous_enable=YES
#
# Uncomment this to allow local users to log in.
# When SELinux is enforcing check for SE bool ftp_home_dir
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
# When SELinux is enforcing check for SE bool allow_ftpd_anon_write, allow_ftpd_full_access
#anon_upload_enable=YES
#
# Uncomment this if you want the anonymous FTP user to be able to create
# new directories.
#anon_mkdir_write_enable=YES
#
# Activate directory messages - messages given to remote users when they
# go into a certain directory.
dirmessage_enable=YES
#
# Activate logging of uploads/downloads.
xferlog_enable=YES
#
# Make sure PORT transfer connections originate from port 20 (ftp-data).
connect_from_port_20=YES
#
# If you want, you can arrange for uploaded anonymous files to be owned by
```

```
# a different user. Note! Using "root" for uploaded files is not
# recommended!
#chown_uploads=YES
#chown_username=whoever
#
# You may override where the log file goes if you like. The default is shown
# below.
#xferlog_file=/var/log/xferlog
#
# If you want, you can have your log file in standard ftpd xferlog format.
# Note that the default log file location is /var/log/xferlog in this case.
xferlog_std_format=YES
#
# You may change the default value for timing out an idle session.
#idle_session_timeout=600
#
# You may change the default value for timing out a data connection.
#data_connection_timeout=120
#
# It is recommended that you define on your system a unique user which the
# ftp server can use as a totally isolated and unprivileged user.
#nopriv_user=ftpsecure
#
# Enable this and the server will recognise asynchronous ABOR requests. Not
# recommended for security (the code is non-trivial). Not enabling it,
# however, may confuse older FTP clients.
#async_abor_enable=YES
#
# By default the server will pretend to allow ASCII mode but in fact ignore
# the request. Turn on the below options to have the server actually do ASCII
# mangling on files when in ASCII mode.
# Beware that on some FTP servers, ASCII support allows a denial of service
# attack (DoS) via the command "SIZE /big/file" in ASCII mode. vsftpd
# predicted this attack and has always been safe, reporting the size of the
```

```
# raw file.
# ASCII mangling is a horrible feature of the protocol.
#ascii_upload_enable=YES
#ascii_download_enable=YES
#
# You may fully customise the login banner string:
#ftpd_banner=Welcome to blah FTP service.
#
# You may specify a file of disallowed anonymous e-mail addresses. Apparently
# useful for combatting certain DoS attacks.
#deny_email_enable=YES
# (default follows)
#banned_email_file=/etc/vsftpd/banned_emails
#
# You may specify an explicit list of local users to chroot() to their home
# directory. If chroot_local_user is YES, then this list becomes a list of
# users to NOT chroot().
# (Warning! chroot'ing can be very dangerous. If using chroot, make sure that
# the user does not have write access to the top level directory within the
# chroot)
#chroot_local_user=YES
#chroot_list_enable=YES
# (default follows)
#chroot_list_file=/etc/vsftpd/chroot_list
#
# You may activate the "-R" option to the builtin ls. This is disabled by
# default to avoid remote users being able to cause excessive I/O on large
# sites. However, some broken FTP clients such as "ncftp" and "mirror" assume
# the presence of the "-R" option, so there is a strong case for enabling it.
#ls_recurse_enable=YES
#
# When "listen" directive is enabled, vsftpd runs in standalone mode and
# listens on IPv4 sockets. This directive cannot be used in conjunction
# with the listen_ipv6 directive.
```

```
listen=NO
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
# Make sure, that one of the listen options is commented !!
listen_ipv6=YES

pam_service_name=vsftpd
userlist_enable=YES
tcp_wrappers=YES
```

Les directives actives de ce fichier sont :

```
[root@linux ~]# grep -E -v '^(#|$)'  /etc/vsftpd/vsftpd.conf > /tmp/vsftpd.conf
[root@linux ~]# cat /etc/vsftpd.conf
cat: /etc/vsftpd.conf: No such file or directory
[root@linux ~]# cat /tmp/vsftpd.conf
anonymous_enable=YES
local_enable=YES
write_enable=YES
local_umask=022
dirmessage_enable=YES
xferlog_enable=YES
connect_from_port_20=YES
xferlog_std_format=YES
listen=NO
listen_ipv6=YES
pam_service_name=vsftpd
userlist_enable=YES
```

```
tcp_wrappers=YES
```

Ces directives sont détaillées ci-après :

Directive	**Valeur par Défaut**	**Description**
anonymous_enable	YES	Si oui, autorise les connexions anonymes
local_enable	YES	Si oui, autorise des connexions par des utilisateurs ayant un compte valide sur le système
write_enable	YES	Si oui, permet l'écriture
local_umask	022	Spécifie la valeur de l'umask lors de la création de fichiers et de répertoires
dirmessage_enable	NO	Si oui, permet d'afficher le contenu du fichier .message quand un utilisateur rentre dans le répertoire
xferlog_enable	NO	Si oui, permet d'activer la journalisation dans le fichier **/var/log/vsftpd.log**
connect_from_port_20	NO	Permet les connexions de ftp-data
listen	NO	Si oui, vfsftpd fonctionne en mode Standalone et non en tant que sous-service de xinetd
pam_service_name	S/O	Indique le nom du service PAM utilisé par vsftpd
userlist_enable	NO	Si oui, vsftpd charge une liste d'utilisateurs spécifiés dans le fichier identifié par la directive **userlist_file**. Si un utilisateur spécifié dans la liste essaie de se connecter, la connexion sera réfusée avant la demande d'un mot de passe
tcp_wrappers	NO	Si oui, vsftpd utilise TCP WRAPPERS

Copiez le fichier **/tmp/vsftpd.conf** vers **/etc/vsftpd/** :

```
[root@linux ~]# cp /etc/vsftpd/vsftpd.conf /root
[root@linux ~]# cp /tmp/vsftpd.conf /etc/vsftpd/vsftpd.conf
cp: overwrite ‘/etc/vsftpd/vsftpd.conf’? y
```

/etc/ftpusers

Votre serveur FTP est maintenant configuré pour les connexions en provenance des utilisateurs ayant un compte sur votre système.

Dans le cas où vous souhaiteriez **interdire** la connexion vers le serveur de certaines personnes mais pas de toutes les personnes ayant un compte système, éditez le fichier **/etc/ftpusers**.

Voici la liste des utilisateurs système qu'il convient d'ajouter à ce fichier:

```
[root@linux ~]# vi /etc/ftpusers
[root@linux ~]# cat /etc/ftpusers
root
bin
daemon
adm
lp
sync
shutdown
halt
mail
uucp
operator
gopher
nobody
dbus
vcsa
rpc
nscd
tcpdump
haldaemon
apache
nslcd
postfix
avahi
ntp
rpcuser
sshd
gdm
vboxadd
named
```

Il est ensuite nécessaire d'inclure une directive supplémentaire à la fin du fichier /etc/vsftpd/vsftpd.conf :

```
[root@linux ~]# vi /etc/vsftpd/vsftpd.conf
[root@linux ~]# cat /etc/vsftpd/vsftpd.conf
anonymous_enable=YES
local_enable=YES
write_enable=YES
local_umask=022
dirmessage_enable=YES
xferlog_enable=YES
connect_from_port_20=YES
xferlog_std_format=YES
listen=NO
listen_ipv6=YES
pam_service_name=vsftpd
userlist_enable=YES
tcp_wrappers=YES
userlist_file=/etc/ftpusers
```

et de redémarrer le serveur :

```
[root@linux ~]# systemctl restart vsftpd
```

Installez le client **ftp** :

```
[root@linux ~]# yum install ftp
```

Testez maintenant le serveur :

```
[root@linux ~]# ftp localhost
Connected to localhost (127.0.0.1).
220 (vsFTPd 3.0.2)
Name (localhost:root): trainee
331 Please specify the password.
```

```
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
257 "/home/trainee"
ftp> exit
```

Bien que trainee puisse se connecter, ce n'est pas le cas pour **root** :

```
[root@linux ~]# ftp localhost
Connected to localhost (127.0.0.1).
220 (vsFTPd 3.0.2)
Name (localhost:root): root
530 Permission denied.
Login failed.
ftp>
```

LAB #4 -Configurer les Plugins de Nagios

Préparation

A Faire : Arrêtez la VM CentOS 7. Dans l'interface de VirtualBox, mettez la VM CentOS 7 dans le Réseau NAT **NatNetwork**. Démarrez ensuite la VM CentOS 7.

Dans la VM CentOS 7, désactivez le parefeu :

```
[root@linux ~]# systemctl stop firewalld
[root@linux ~]# systemctl disable firewalld
```

Mettez SELINUX en mode **permissive** :

```
[root@linux ~]# setenforce permissive
[root@linux ~]# vi /etc/sysconfig/selinux
[root@linux ~]# cat /etc/sysconfig/selinux

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of three two values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

A Faire : Retournez à votre VM Debian 8 et ouvrez une session en tant que root en utilisant la commande **su -**.

Le Serveur Web Apache

Créez le fichier **/usr/local/nagios/etc/services/linux-www.cfg** pour surveiller le serveur web de notre hôte Nagios :

```
root@nagios:~# vi /usr/local/nagios/etc/services/linux-www.cfg
root@nagios:~# cat /usr/local/nagios/etc/services/linux-www.cfg
  define service{
    host_name                       linux
    service_description             www
    check_command                   check_http
```

```
    check_interval                10
    check_period                  24x7
    retry_interval                3
    max_check_attempts            3
    notification_interval         30
    notification_period           24x7
    notification_options          w,c,u,r
    contact_groups                admins
    }
```

La syntaxe du plugin Nagios **check_http** est :

```
root@nagios:~# /usr/local/nagios/libexec/check_http --help
check_http v2.1.2 (nagios-plugins 2.1.2)
Copyright (c) 1999 Ethan Galstad <nagios@nagios.org>
Copyright (c) 1999-2014 Nagios Plugin Development Team
    <devel@nagios-plugins.org>

This plugin tests the HTTP service on the specified host. It can test
normal (http) and secure (https) servers, follow redirects, search for
strings and regular expressions, check connection times, and report on
certificate expiration times.


Usage:
 check_http -H <vhost> | -I <IP-address> [-u <uri>] [-p <port>]
       [-J <client certificate file>] [-K <private key>]
       [-w <warn time>] [-c <critical time>] [-t <timeout>] [-L] [-E] [-a auth]
       [-b proxy_auth] [-f <ok|warning|critcal|follow|sticky|stickyport>]
       [-e <expect>] [-d string] [-s string] [-l] [-r <regex> | -R <case-insensitive regex>]
       [-P string] [-m <min_pg_size>:<max_pg_size>] [-4|-6] [-N] [-M <age>]
       [-A string] [-k string] [-S <version>] [--sni] [-C <warn_age>[,<crit_age>]]
       [-T <content-type>] [-j method]
NOTE: One or both of -H and -I must be specified
```

```
Options:
 -h, --help
    Print detailed help screen
 -V, --version
    Print version information
 --extra-opts=[section][@file]
    Read options from an ini file. See
    https://www.nagios-plugins.org/doc/extra-opts.html
    for usage and examples.
 -H, --hostname=ADDRESS
    Host name argument for servers using host headers (virtual host)
    Append a port to include it in the header (eg: example.com:5000)
 -I, --IP-address=ADDRESS
    IP address or name (use numeric address if possible to bypass DNS lookup).
 -p, --port=INTEGER
    Port number (default: 80)
 -4, --use-ipv4
    Use IPv4 connection
 -6, --use-ipv6
    Use IPv6 connection
 -S, --ssl=VERSION[+]
    Connect via SSL. Port defaults to 443. VERSION is optional, and prevents
    auto-negotiation (2 = SSLv2, 3 = SSLv3, 1 = TLSv1, 1.1 = TLSv1.1,
    1.2 = TLSv1.2). With a '+' suffix, newer versions are also accepted.
 --sni
    Enable SSL/TLS hostname extension support (SNI)
 -C, --certificate=INTEGER[,INTEGER]
    Minimum number of days a certificate has to be valid. Port defaults to 443
    (when this option is used the URL is not checked.)
 -J, --client-cert=FILE
   Name of file that contains the client certificate (PEM format)
   to be used in establishing the SSL session
 -K, --private-key=FILE
   Name of file containing the private key (PEM format)
```

```
  matching the client certificate
-e, --expect=STRING
   Comma-delimited list of strings, at least one of them is expected in
   the first (status) line of the server response (default: HTTP/1.)
   If specified skips all other status line logic (ex: 3xx, 4xx, 5xx processing)
-d, --header-string=STRING
   String to expect in the response headers
-s, --string=STRING
   String to expect in the content
-u, --url=PATH
   URL to GET or POST (default: /)
-P, --post=STRING
   URL encoded http POST data
-j, --method=STRING  (for example: HEAD, OPTIONS, TRACE, PUT, DELETE, CONNECT)
   Set HTTP method.
-N, --no-body
   Don't wait for document body: stop reading after headers.
   (Note that this still does an HTTP GET or POST, not a HEAD.)
-M, --max-age=SECONDS
   Warn if document is more than SECONDS old. the number can also be of
   the form "10m" for minutes, "10h" for hours, or "10d" for days.
-T, --content-type=STRING
   specify Content-Type header media type when POSTing

-l, --linespan
   Allow regex to span newlines (must precede -r or -R)
-r, --regex, --ereg=STRING
   Search page for regex STRING
-R, --eregi=STRING
   Search page for case-insensitive regex STRING
--invert-regex
   Return CRITICAL if found, OK if not

-a, --authorization=AUTH_PAIR
```

```
   Username:password on sites with basic authentication
 -b, --proxy-authorization=AUTH_PAIR
   Username:password on proxy-servers with basic authentication
 -A, --useragent=STRING
   String to be sent in http header as "User Agent"
 -k, --header=STRING
   Any other tags to be sent in http header. Use multiple times for additional headers
 -E, --extended-perfdata
   Print additional performance data
 -L, --link
   Wrap output in HTML link (obsoleted by urlize)
 -f, --onredirect=<ok|warning|critical|follow|sticky|stickyport>
   How to handle redirected pages. sticky is like follow but stick to the
   specified IP address. stickyport also ensures port stays the same.
 -m, --pagesize=INTEGER<:INTEGER>
   Minimum page size required (bytes) : Maximum page size required (bytes)
 -w, --warning=DOUBLE
   Response time to result in warning status (seconds)
 -c, --critical=DOUBLE
   Response time to result in critical status (seconds)
 -t, --timeout=INTEGER:<timeout state>
   Seconds before connection times out (default: 10)
   Optional ":<timeout state>" can be a state integer (0,1,2,3) or a state STRING
 -v, --verbose
   Show details for command-line debugging (Nagios may truncate output)

Notes:
 This plugin will attempt to open an HTTP connection with the host.
 Successful connects return STATE_OK, refusals and timeouts return STATE_CRITICAL
 other errors return STATE_UNKNOWN.  Successful connects, but incorrect reponse
 messages from the host result in STATE_WARNING return values.  If you are
 checking a virtual server that uses 'host headers' you must supply the FQDN
 (fully qualified domain name) as the [host_name] argument.
```

```
 This plugin can also check whether an SSL enabled web server is able to
 serve content (optionally within a specified time) or whether the X509
 certificate is still valid for the specified number of days.

 Please note that this plugin does not check if the presented server
 certificate matches the hostname of the server, or if the certificate
 has a valid chain of trust to one of the locally installed CAs.

Examples:
 CHECK CONTENT: check_http -w 5 -c 10 --ssl -H www.verisign.com

 When the 'www.verisign.com' server returns its content within 5 seconds,
 a STATE_OK will be returned. When the server returns its content but exceeds
 the 5-second threshold, a STATE_WARNING will be returned. When an error occurs,
 a STATE_CRITICAL will be returned.

 CHECK CERTIFICATE: check_http -H www.verisign.com -C 14

 When the certificate of 'www.verisign.com' is valid for more than 14 days,
 a STATE_OK is returned. When the certificate is still valid, but for less than
 14 days, a STATE_WARNING is returned. A STATE_CRITICAL will be returned when
 the certificate is expired.


 CHECK CERTIFICATE: check_http -H www.verisign.com -C 30,14

 When the certificate of 'www.verisign.com' is valid for more than 30 days,
 a STATE_OK is returned. When the certificate is still valid, but for less than
 30 days, but more than 14 days, a STATE_WARNING is returned.
 A STATE_CRITICAL will be returned when certificate expires in less than 14 days
 CHECK SSL WEBSERVER CONTENT VIA PROXY USING HTTP 1.1 CONNECT:

 check_http -I 192.168.100.35 -p 80 -u https://www.verisign.com/ -S -j CONNECT -H www.verisign.com
 all these options are needed: -I <proxy> -p <proxy-port> -u <check-url> -S(sl) -j CONNECT -H <webserver>
```

```
 a STATE_OK will be returned. When the server returns its content but exceeds
 the 5-second threshold, a STATE_WARNING will be returned. When an error occurs,
 a STATE_CRITICAL will be returned.

Send email to help@nagios-plugins.org if you have questions regarding use
of this software. To submit patches or suggest improvements, send email to
devel@nagios-plugins.org
```

Vérifiez la configuration de Nagios :

```
root@nagios:~# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.2.0
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 08-01-2016
License: GPL

Website: https://www.nagios.org
Reading configuration data...
   Read main config file okay...
   Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
    Checked 5 services.
    Checked 2 hosts.
    Checked 0 host groups.
    Checked 0 service groups.
    Checked 1 contacts.
    Checked 2 contact groups.
    Checked 25 commands.
    Checked 3 time periods.
```

```
    Checked 0 host escalations.
    Checked 0 service escalations.
Checking for circular paths...
    Checked 2 hosts
    Checked 0 service dependencies
    Checked 0 host dependencies
    Checked 3 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors:   0

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

Redémarrez le service nagios :

```
root@nagios:~# systemctl restart nagios
root@nagios:~# systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
   Loaded: loaded (/etc/init.d/nagios)
   Active: active (running) since Fri 2018-05-18 11:02:49 BST; 8s ago
  Process: 5578 ExecStop=/etc/init.d/nagios stop (code=exited, status=0/SUCCESS)
  Process: 5585 ExecStart=/etc/init.d/nagios start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/nagios.service
           ├─5605 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─5607 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─5608 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─5609 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─5610 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─5611 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

May 18 11:02:49 nagios.i2tch.loc nagios[5605]: nerd: Channel hostchecks registered successfully
```

```
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: nerd: Channel servicechecks registered successfully
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: nerd: Channel opathchecks registered successfully
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: nerd: Fully initialized and ready to rock!
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: wproc: Successfully registered manager as @wpro...ler
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: wproc: Registry request: name=Core Worker 5609;...609
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: wproc: Registry request: name=Core Worker 5610;...610
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: wproc: Registry request: name=Core Worker 5608;...608
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: wproc: Registry request: name=Core Worker 5607;...607
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: Successfully launched command file worker with ...611
Hint: Some lines were ellipsized, use -l to show in full.
```

Lors de la vérification suivante par Nagios, l'état du service **www** devrait être **OK** :

```
root@nagios:~# ncmd -q list -h linux
Fetching services and health on linux
---
Service  State
---
ssh      OK
www      OK
```

Le Serveur de Bases de Données MariaDB

Nagios fournit deux plugins pour MySQL / MariaDB :

- **check-mysql** - la vérification de la connectivité à la base de données ainsi que l'état de la réplication,
- **check_mysql_query** - la mesure du temps d'exécution d'une requête SQL.

La syntaxe du plugin **check-mysql** est :

```
root@nagios:~# /usr/local/nagios/libexec/check_mysql --help
check_mysql v2.1.2 (nagios-plugins 2.1.2)
Copyright (c) 1999-2014 Nagios Plugin Development Team
```

```
    <devel@nagios-plugins.org>

This program tests connections to a MySQL server


Usage:
 check_mysql [-d database] [-H host] [-P port] [-s socket]
       [-u user] [-p password] [-S] [-l] [-a cert] [-k key]
       [-C ca-cert] [-D ca-dir] [-L ciphers] [-f optfile] [-g group]

Options:
 -h, --help
    Print detailed help screen
 -V, --version
    Print version information
 --extra-opts=[section][@file]
    Read options from an ini file. See
    https://www.nagios-plugins.org/doc/extra-opts.html
    for usage and examples.
 -H, --hostname=ADDRESS
    Host name, IP Address, or unix socket (must be an absolute path)
 -P, --port=INTEGER
    Port number (default: 3306)
 -n, --ignore-auth
    Ignore authentication failure and check for mysql connectivity only
 -s, --socket=STRING
    Use the specified socket (has no effect if -H is used)
 -d, --database=STRING
    Check database with indicated name
 -f, --file=STRING
    Read from the specified client options file
 -g, --group=STRING
    Use a client options group
 -u, --username=STRING
```

```
   Connect using the indicated username
 -p, --password=STRING
   Use the indicated password to authenticate the connection
   ==> IMPORTANT: THIS FORM OF AUTHENTICATION IS NOT SECURE!!! <==
   Your clear-text password could be visible as a process table entry
 -S, --check-slave
   Check if the slave thread is running properly.
 -w, --warning
   Exit with WARNING status if slave server is more than INTEGER seconds
   behind master
 -c, --critical
   Exit with CRITICAL status if slave server is more then INTEGER seconds
   behind master
 -l, --ssl
   Use ssl encryptation
 -C, --ca-cert=STRING
   Path to CA signing the cert
 -a, --cert=STRING
   Path to SSL certificate
 -k, --key=STRING
   Path to private SSL key
 -D, --ca-dir=STRING
   Path to CA directory
 -L, --ciphers=STRING
   List of valid SSL ciphers

 There are no required arguments. By default, the local database is checked
 using the default unix socket. You can force TCP on localhost by using an
 IP address or FQDN ('localhost' will use the socket as well).

Notes:
 You must specify -p with an empty string to force an empty password,
 overriding any my.cnf settings.
```

```
Send email to help@nagios-plugins.org if you have questions regarding use
of this software. To submit patches or suggest improvements, send email to
devel@nagios-plugins.org
```

La syntaxe du plugin **check_mysql_query** est :

```
root@nagios:~# /usr/local/nagios/libexec/check_mysql_query --help
check_mysql_query v2.1.2 (nagios-plugins 2.1.2)
Copyright (c) 1999-2014 Nagios Plugin Development Team
    <devel@nagios-plugins.org>

This program checks a query result against threshold levels



Usage:
 check_mysql_query -q SQL_query [-w warn] [-c crit] [-H host] [-P port] [-s socket]
       [-d database] [-u user] [-p password] [-f optfile] [-g group]

Options:
 -h, --help
    Print detailed help screen
 -V, --version
    Print version information
 --extra-opts=[section][@file]
    Read options from an ini file. See
    https://www.nagios-plugins.org/doc/extra-opts.html
    for usage and examples.
 -q, --query=STRING
    SQL query to run. Only first column in first row will be read
 -w, --warning=RANGE
    Warning range (format: start:end). Alert if outside this range
 -c, --critical=RANGE
    Critical range
 -H, --hostname=ADDRESS
```

```
   Host name, IP Address, or unix socket (must be an absolute path)
 -P, --port=INTEGER
   Port number (default: 3306)
 -s, --socket=STRING
   Use the specified socket (has no effect if -H is used)
 -d, --database=STRING
   Database to check
 -f, --file=STRING
   Read from the specified client options file
 -g, --group=STRING
   Use a client options group
 -u, --username=STRING
   Username to login with
 -p, --password=STRING
   Password to login with
   ==> IMPORTANT: THIS FORM OF AUTHENTICATION IS NOT SECURE!!! <==
   Your clear-text password could be visible as a process table entry

 A query is required. The result from the query should be numeric.
 For extra security, create a user with minimal access.

Notes:
 You must specify -p with an empty string to force an empty password,
 overriding any my.cnf settings.

Send email to help@nagios-plugins.org if you have questions regarding use
of this software. To submit patches or suggest improvements, send email to
devel@nagios-plugins.org
```

Créez le fichier de commandes suivant :

```
root@nagios:~# vi /usr/local/nagios/etc/commands/mysql.cfg
root@nagios:~# cat /usr/local/nagios/etc/commands/mysql.cfg
define command{
```

```
    command_name  check_mysql
    command_line  $USER1$/check_mysql -H $HOSTADDRESS$ -u nagios -p password -d nagios -w 10 -c 30
  }
```

Créez ensuite le fichier **/usr/local/nagios/etc/services/linux-mysql.cfg** pour surveiller le serveur MariaDB :

```
root@nagios:~# vi /usr/local/nagios/etc/services/linux-mysql.cfg
root@nagios:~# cat /usr/local/nagios/etc/services/linux-mysql.cfg
  define service{
    host_name                       linux
    service_description             mariadb
    check_command                   check_mysql
    check_interval                  5
    retry_interval                  1
    max_check_attempts              3
    check_period                    24x7
    contact_groups                  admins
    notification_interval           60
    notification_period             24x7
    notification_options            w,c,u,r
    }
```

Re-démarrez le service nagios :

```
root@nagios:~# systemctl restart nagios
root@nagios:~# systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
   Loaded: loaded (/etc/init.d/nagios)
   Active: active (running) since Fri 2018-05-18 11:02:49 BST; 8s ago
  Process: 5578 ExecStop=/etc/init.d/nagios stop (code=exited, status=0/SUCCESS)
  Process: 5585 ExecStart=/etc/init.d/nagios start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/nagios.service
           ├─5605 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─5607 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
```

```
           ├─5608 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─5609 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─5610 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─5611 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

May 18 11:02:49 nagios.i2tch.loc nagios[5605]: nerd: Channel hostchecks registered successfully
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: nerd: Channel servicechecks registered successfully
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: nerd: Channel opathchecks registered successfully
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: nerd: Fully initialized and ready to rock!
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: wproc: Successfully registered manager as @wpro...ler
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: wproc: Registry request: name=Core Worker 5609;...609
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: wproc: Registry request: name=Core Worker 5610;...610
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: wproc: Registry request: name=Core Worker 5608;...608
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: wproc: Registry request: name=Core Worker 5607;...607
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: Successfully launched command file worker with ...611
Hint: Some lines were ellipsized, use -l to show in full.
```

Lors de la vérification suivante par Nagios, l'état du service **mariadb** devrait être **OK** :

```
root@nagios:~# ncmd -q list -h linux
Fetching services and health on linux
---
Service  State
---
mariadb  OK
ssh      OK
www      OK
```

Le Serveur SMTP Postfix

Nagios fournit trois plugins pour les serveurs mail :

- **check_pop3** - la vérification du serveur POP3,

- **check_imap** - la vérification du serveur IMAP,
- **check_smtp** - la vérification du serveur SMTP.

La syntaxe du plugin **check-smtp** est :

```
root@nagios:~# /usr/local/nagios/libexec/check_smtp --help
check_smtp v2.1.2 (nagios-plugins 2.1.2)
Copyright (c) 1999-2001 Ethan Galstad <nagios@nagios.org>
Copyright (c) 2000-2014 Nagios Plugin Development Team
    <devel@nagios-plugins.org>

This plugin will attempt to open an SMTP connection with the host.


Usage:
check_smtp -H host [-p port] [-4|-6] [-e expect] [-C command] [-R response] [-f from addr]
[-A authtype -U authuser -P authpass] [-w warn] [-c crit] [-t timeout] [-q]
[-F fqdn] [-S] [-D warn days cert expire[,crit days cert expire]] [-v]

Options:
 -h, --help
    Print detailed help screen
 -V, --version
    Print version information
 --extra-opts=[section][@file]
    Read options from an ini file. See
    https://www.nagios-plugins.org/doc/extra-opts.html
    for usage and examples.
 -H, --hostname=ADDRESS
    Host name, IP Address, or unix socket (must be an absolute path)
 -p, --port=INTEGER
    Port number (default: 25)
 -4, --use-ipv4
    Use IPv4 connection
```

```
-6, --use-ipv6
   Use IPv6 connection
-e, --expect=STRING
   String to expect in first line of server response (default: '220')
-C, --command=STRING
   SMTP command (may be used repeatedly)
-R, --response=STRING
   Expected response to command (may be used repeatedly)
-f, --from=STRING
   FROM-address to include in MAIL command, required by Exchange 2000
-F, --fqdn=STRING
   FQDN used for HELO
-D, --certificate=INTEGER[,INTEGER]
   Minimum number of days a certificate has to be valid.
-S, --starttls
   Use STARTTLS for the connection.
-A, --authtype=STRING
   SMTP AUTH type to check (default none, only LOGIN supported)
-U, --authuser=STRING
   SMTP AUTH username
-P, --authpass=STRING
   SMTP AUTH password
-q, --ignore-quit-failure
   Ignore failure when sending QUIT command to server
-w, --warning=DOUBLE
   Response time to result in warning status (seconds)
-c, --critical=DOUBLE
   Response time to result in critical status (seconds)
-t, --timeout=INTEGER:<timeout state>
   Seconds before connection times out (default: 10)
   Optional ":<timeout state>" can be a state integer (0,1,2,3) or a state STRING
-v, --verbose
   Show details for command-line debugging (Nagios may truncate output)
```

```
Successul connects return STATE_OK, refusals and timeouts return
STATE_CRITICAL, other errors return STATE_UNKNOWN.  Successful
connects, but incorrect reponse messages from the host result in
STATE_WARNING return values.

Send email to help@nagios-plugins.org if you have questions regarding use
of this software. To submit patches or suggest improvements, send email to
devel@nagios-plugins.org
```

Créez le fichier de commandes suivant :

```
root@nagios:~# vi /usr/local/nagios/etc/commands/postfix.cfg
root@nagios:~# cat /usr/local/nagios/etc/commands/postfix.cfg
define command{
    command_name  check_smtpsend
    command_line  $USER1$/check_smtp -H $HOSTADDRESS$ -f trainee@i2tch.com -C "RCPT TO: postmaster@i2tch.com" -R
"250"
  }
```

Créez ensuite le fichier **/usr/local/nagios/etc/services/linux-postfix.cfg** pour surveiller le serveur postfix :

```
root@nagios:~# vi /usr/local/nagios/etc/services/linux-postfix.cfg
root@nagios:~# cat /usr/local/nagios/etc/services/linux-postfix.cfg
  define service{
    host_name                       linux
    service_description             postfix
    check_command                   check_smtpsend
    check_interval                  5
    retry_interval                  1
    max_check_attempts              3
    check_period                    24x7
    contact_groups                  admins
    notification_interval           60
    notification_period             24x7
```

```
    notification_options            w,c,u,r
    }
```

Vérifiez la configuration de Nagios :

```
root@nagios:~# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.2.0
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 08-01-2016
License: GPL

Website: https://www.nagios.org
Reading configuration data...
   Read main config file okay...
   Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
    Checked 6 services.
    Checked 2 hosts.
    Checked 0 host groups.
    Checked 0 service groups.
    Checked 1 contacts.
    Checked 2 contact groups.
    Checked 26 commands.
    Checked 3 time periods.
    Checked 0 host escalations.
    Checked 0 service escalations.
Checking for circular paths...
    Checked 2 hosts
    Checked 0 service dependencies
```

```
    Checked 0 host dependencies
    Checked 3 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors:   0

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

Re-démarrez le service nagios :

```
root@nagios:~# systemctl restart nagios
root@nagios:~# systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
   Loaded: loaded (/etc/init.d/nagios)
   Active: active (running) since Fri 2018-05-18 11:02:49 BST; 8s ago
  Process: 5578 ExecStop=/etc/init.d/nagios stop (code=exited, status=0/SUCCESS)
  Process: 5585 ExecStart=/etc/init.d/nagios start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/nagios.service
           ├─5605 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─5607 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─5608 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─5609 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─5610 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─5611 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

May 18 11:02:49 nagios.i2tch.loc nagios[5605]: nerd: Channel hostchecks registered successfully
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: nerd: Channel servicechecks registered successfully
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: nerd: Channel opathchecks registered successfully
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: nerd: Fully initialized and ready to rock!
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: wproc: Successfully registered manager as @wpro...ler
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: wproc: Registry request: name=Core Worker 5609;...609
```

```
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: wproc: Registry request: name=Core Worker 5610;...610
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: wproc: Registry request: name=Core Worker 5608;...608
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: wproc: Registry request: name=Core Worker 5607;...607
May 18 11:02:49 nagios.i2tch.loc nagios[5605]: Successfully launched command file worker with ...611
Hint: Some lines were ellipsized, use -l to show in full.
```

Lors de la vérification suivante par Nagios, l'état du service **postfix** devrait être **OK** :

```
root@nagios:~# ncmd -q list -h linux
Fetching services and health on linux
---
Service  State
---
mariadb  OK
postfix  OK
ssh      OK
www      OK
```

Le Serveur FTP vsftpd

Nagios fournit un plugin pour les serveurs FTP :

- **check-ftp** - la vérification du serveur FTP.

La syntaxe du plugin **check-ftp** est :

```
root@nagios:~# /usr/local/nagios/libexec/check_ftp --help
check_ftp v2.1.2 (nagios-plugins 2.1.2)
Copyright (c) 1999 Ethan Galstad <nagios@nagios.org>
Copyright (c) 1999-2014 Nagios Plugin Development Team
    <devel@nagios-plugins.org>

This plugin tests FTP connections with the specified host (or unix socket).
```

```
Usage:
check_ftp -H host -p port [-w <warning time>] [-c <critical time>] [-s <send string>]
[-e <expect string>] [-q <quit string>][-m <maximum bytes>] [-d <delay>]
[-t <timeout seconds>] [-r <refuse state>] [-M <mismatch state>] [-v] [-4|-6] [-j]
[-D <warn days cert expire>[,<crit days cert expire>]] [-S <use SSL>] [-E]

Options:
 -h, --help
    Print detailed help screen
 -V, --version
    Print version information
 --extra-opts=[section][@file]
    Read options from an ini file. See
    https://www.nagios-plugins.org/doc/extra-opts.html
    for usage and examples.
 -H, --hostname=ADDRESS
    Host name, IP Address, or unix socket (must be an absolute path)
 -p, --port=INTEGER
    Port number (default: none)
 -4, --use-ipv4
    Use IPv4 connection
 -6, --use-ipv6
    Use IPv6 connection
 -E, --escape
    Can use \n, \r, \t or \\ in send or quit string. Must come before send or quit option
    Default: nothing added to send, \r\n added to end of quit
 -s, --send=STRING
    String to send to the server
 -e, --expect=STRING
    String to expect in server response (may be repeated)
 -A, --all
    All expect strings need to occur in server response. Default is any
 -q, --quit=STRING
    String to send server to initiate a clean close of the connection
```

```
 -r, --refuse=ok|warn|crit
    Accept TCP refusals with states ok, warn, crit (default: crit)
 -M, --mismatch=ok|warn|crit
    Accept expected string mismatches with states ok, warn, crit (default: warn)
 -j, --jail
    Hide output from TCP socket
 -m, --maxbytes=INTEGER
    Close connection once more than this number of bytes are received
 -d, --delay=INTEGER
    Seconds to wait between sending string and polling for response
 -D, --certificate=INTEGER[,INTEGER]
    Minimum number of days a certificate has to be valid.
    1st is #days for warning, 2nd is critical (if not specified - 0).
 -S, --ssl
    Use SSL for the connection.
 -w, --warning=DOUBLE
    Response time to result in warning status (seconds)
 -c, --critical=DOUBLE
    Response time to result in critical status (seconds)
 -t, --timeout=INTEGER:<timeout state>
    Seconds before connection times out (default: 10)
    Optional ":<timeout state>" can be a state integer (0,1,2,3) or a state STRING
 -v, --verbose
    Show details for command-line debugging (Nagios may truncate output)

Send email to help@nagios-plugins.org if you have questions regarding use
of this software. To submit patches or suggest improvements, send email to
devel@nagios-plugins.org
```

Créez le fichier de commandes suivant :

```
root@nagios:~# vi /usr/local/nagios/etc/commands/vsftpd.cfg
root@nagios:~# cat /usr/local/nagios/etc/commands/vsftpd.cfg
 define command{
```

```
    command_name  check_ftplogin
    command_line  $USER1$/check_ftp -H $HOSTADDRESS$ -E -s "USER trainee\r\nPASS trainee\r\n" -d 5 -e "230"
  }
```

Créez ensuite le fichier **/usr/local/nagios/etc/services/linux-vsftpd.cfg** pour surveiller le serveur postfix :

```
root@nagios:~# vi /usr/local/nagios/etc/services/linux-vsftpd.cfg
root@nagios:~# cat /usr/local/nagios/etc/services/linux-vsftpd.cfg
  define service{
    host_name                       linux
    service_description             vsftpd
    check_command                   check_ftplogin
    check_interval                  5
    retry_interval                  1
    max_check_attempts              3
    check_period                    24x7
    contact_groups                  admins
    notification_interval           60
    notification_period             24x7
    notification_options            w,c,u,r
    }
```

Vérifiez la configuration de Nagios :

```
root@nagios:~# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.2.0
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 08-01-2016
License: GPL

Website: https://www.nagios.org
Reading configuration data...
```

```
   Read main config file okay...
   Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
    Checked 7 services.
    Checked 2 hosts.
    Checked 0 host groups.
    Checked 0 service groups.
    Checked 1 contacts.
    Checked 2 contact groups.
    Checked 27 commands.
    Checked 3 time periods.
    Checked 0 host escalations.
    Checked 0 service escalations.
Checking for circular paths...
    Checked 2 hosts
    Checked 0 service dependencies
    Checked 0 host dependencies
    Checked 3 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors:   0

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

Re-démarrez le service nagios :

```
root@nagios:~# systemctl restart nagios
root@nagios:~# systemctl status nagios
```

```
● nagios.service - LSB: Starts and stops the Nagios monitoring server
   Loaded: loaded (/etc/init.d/nagios)
   Active: active (running) since Sun 2018-05-20 10:29:23 BST; 6s ago
  Process: 20149 ExecStop=/etc/init.d/nagios stop (code=exited, status=0/SUCCESS)
  Process: 20156 ExecStart=/etc/init.d/nagios start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/nagios.service
           ├─20176 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─20178 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─20179 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─20180 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─20181 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─20182 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

May 20 10:29:23 nagios.i2tch.loc nagios[20176]: nerd: Channel servicechecks registered successfully
May 20 10:29:23 nagios.i2tch.loc nagios[20176]: nerd: Channel opathchecks registered successfully
May 20 10:29:23 nagios.i2tch.loc nagios[20176]: nerd: Fully initialized and ready to rock!
May 20 10:29:23 nagios.i2tch.loc nagios[20176]: wproc: Successfully registered manager as @wproc with query
handler
May 20 10:29:23 nagios.i2tch.loc nagios[20176]: wproc: Registry request: name=Core Worker 20181;pid=20181
May 20 10:29:23 nagios.i2tch.loc nagios[20176]: wproc: Registry request: name=Core Worker 20180;pid=20180
May 20 10:29:23 nagios.i2tch.loc nagios[20176]: wproc: Registry request: name=Core Worker 20179;pid=20179
May 20 10:29:23 nagios.i2tch.loc nagios[20176]: wproc: Registry request: name=Core Worker 20178;pid=20178
May 20 10:29:23 nagios.i2tch.loc nagios[20176]: SERVICE FLAPPING ALERT: linux;www;STARTED; Service appears to
have started fl...shold)
May 20 10:29:23 nagios.i2tch.loc nagios[20176]: Successfully launched command file worker with pid 20182
Hint: Some lines were ellipsized, use -l to show in full.
```

Lors de la vérification suivante par Nagios, l'état du service **ftp** devrait être **OK** :

```
root@nagios:~# ncmd -q list -h linux
Fetching services and health on linux
---
Service  State
---
```

```
mariadb  OK
postfix  OK
ssh      OK
vsftpd   OK
www      OK
```

LAB #5 -Vérifications Supplémentaires sur le Serveur Nagios

L'Espace Disque

Nagios fournit plusieurs plugins pour la surveillance de l'espace disque :

- **check_swap** - la surveillance de l'espace swap disponible,
- **check_disk** - la surveillance de l'espace disque.

La syntaxe du plugin **check_swap** est :

```
root@nagios:~# /usr/local/nagios/libexec/check_swap --help
check_swap v2.1.2 (nagios-plugins 2.1.2)
Copyright (c) 2000-2014 Nagios Plugin Development Team
    <devel@nagios-plugins.org>

Check swap space on local machine.


Usage:
 check_swap [-av] -w <percent_free>% -c <percent_free>%
  -w <bytes_free> -c <bytes_free>

Options:
 -h, --help
    Print detailed help screen
```

```
 -V, --version
    Print version information
 --extra-opts=[section][@file]
    Read options from an ini file. See
    https://www.nagios-plugins.org/doc/extra-opts.html
    for usage and examples.
 -w, --warning=INTEGER
    Exit with WARNING status if less than INTEGER bytes of swap space are free
 -w, --warning=PERCENT%%
    Exit with WARNING status if less than PERCENT of swap space is free
 -c, --critical=INTEGER
    Exit with CRITICAL status if less than INTEGER bytes of swap space are free
 -c, --critical=PERCENT%%
    Exit with CRITICAL status if less than PERCENT of swap space is free
 -a, --allswaps
    Conduct comparisons for all swap partitions, one by one
 -v, --verbose
    Show details for command-line debugging (Nagios may truncate output)

Notes:
 Both INTEGER and PERCENT thresholds can be specified, they are all checked.
 On AIX, if -a is specified, uses lsps -a, otherwise uses lsps -s.

Send email to help@nagios-plugins.org if you have questions regarding use
of this software. To submit patches or suggest improvements, send email to
devel@nagios-plugins.org
```

La syntaxe du plugin **check_disk** est :

```
root@nagios:~# /usr/local/nagios/libexec/check_disk --help
check_disk v2.1.2 (nagios-plugins 2.1.2)
Copyright (c) 1999 Ethan Galstad <nagios@nagios.org>
Copyright (c) 1999-2014 Nagios Plugin Development Team
    <devel@nagios-plugins.org>
```

```
This plugin checks the amount of used disk space on a mounted file system
and generates an alert if free space is less than one of the threshold values



Usage:
 check_disk -w limit -c limit [-W limit] [-K limit] {-p path | -x device}
[-C] [-E] [-e] [-f] [-g group ] [-k] [-l] [-M] [-m] [-R path ] [-r path ]
[-t timeout] [-u unit] [-v] [-X type] [-N type] [-n]

Options:
 -h, --help
    Print detailed help screen
 -V, --version
    Print version information
 --extra-opts=[section][@file]
    Read options from an ini file. See
    https://www.nagios-plugins.org/doc/extra-opts.html
    for usage and examples.
 -w, --warning=INTEGER
    Exit with WARNING status if less than INTEGER units of disk are free
 -w, --warning=PERCENT%
    Exit with WARNING status if less than PERCENT of disk space is free
 -c, --critical=INTEGER
    Exit with CRITICAL status if less than INTEGER units of disk are free
 -c, --critical=PERCENT%
    Exit with CRITICAL status if less than PERCENT of disk space is free
 -W, --iwarning=PERCENT%
    Exit with WARNING status if less than PERCENT of inode space is free
 -K, --icritical=PERCENT%
    Exit with CRITICAL status if less than PERCENT of inode space is free
 -p, --path=PATH, --partition=PARTITION
    Mount point or block device as emitted by the mount(8) command (may be repeated)
 -x, --exclude_device=PATH <STRING>
    Ignore device (only works if -p unspecified)
```

```
-C, --clear
   Clear thresholds
-E, --exact-match
   For paths or partitions specified with -p, only check for exact paths
-e, --errors-only
   Display only devices/mountpoints with errors
-f, --freespace-ignore-reserved
   Don't account root-reserved blocks into freespace in perfdata
-g, --group=NAME
   Group paths. Thresholds apply to (free-)space of all partitions together
-k, --kilobytes
   Same as '--units kB'
-l, --local
   Only check local filesystems
-L, --stat-remote-fs
   Only check local filesystems against thresholds. Yet call stat on remote filesystems
   to test if they are accessible (e.g. to detect Stale NFS Handles)
-M, --mountpoint
   Display the mountpoint instead of the partition
-m, --megabytes
   Same as '--units MB'
-A, --all
   Explicitly select all paths. This is equivalent to -R '.*'
-R, --eregi-path=PATH, --eregi-partition=PARTITION
   Case insensitive regular expression for path/partition (may be repeated)
-r, --ereg-path=PATH, --ereg-partition=PARTITION
   Regular expression for path or partition (may be repeated)
-I, --ignore-eregi-path=PATH, --ignore-eregi-partition=PARTITION
   Regular expression to ignore selected path/partition (case insensitive) (may be repeated)
-i, --ignore-ereg-path=PATH, --ignore-ereg-partition=PARTITION
   Regular expression to ignore selected path or partition (may be repeated)
-t, --timeout=INTEGER
   Seconds before plugin times out (default: 10)
-u, --units=STRING
```

```
    Choose bytes, kB, MB, GB, TB (default: MB)
 -v, --verbose
    Show details for command-line debugging (Nagios may truncate output)
 -X, --exclude-type=TYPE
    Ignore all filesystems of indicated type (may be repeated)
 -N, --include-type=TYPE
    Check only filesystems of indicated type (may be repeated)
 -n, --newlines
    Multi-line output of each disk's status information on a new line

Examples:
 check_disk -w 10% -c 5% -p /tmp -p /var -C -w 100000 -c 50000 -p /
    Checks /tmp and /var at 10% and 5%, and / at 100MB and 50MB
 check_disk -w 100 -c 50 -C -w 1000 -c 500 -g sidDATA -r '^/oracle/SID/data.*$'
    Checks all filesystems not matching -r at 100M and 50M. The fs matching the -r regex
    are grouped which means the freespace thresholds are applied to all disks together
 check_disk -w 100 -c 50 -C -w 1000 -c 500 -p /foo -C -w 5% -c 3% -p /bar
    Checks /foo for 1000M/500M and /bar for 5/3%. All remaining volumes use 100M/50M

Send email to help@nagios-plugins.org if you have questions regarding use
of this software. To submit patches or suggest improvements, send email to
devel@nagios-plugins.org
```

Créez le fichier de commandes suivant :

```
root@nagios:~# vi /usr/local/nagios/etc/commands/disks.cfg
root@nagios:~# cat /usr/local/nagios/etc/commands/disks.cfg
 define command{
    command_name  check_swap
    command_line  $USER1$/check_swap -w 20% -c 10%
  }

define command{
    command_name  check_partition
```

```
    command_line  $USER1$/check_disk -w 10% -c 5% -A -e
  }
```

Créez ensuite les fichiers suivants :

- **/usr/local/nagios/etc/services/localhost-swap.cfg**,
- **/usr/local/nagios/etc/services/localhost-disk.cfg**.

```
root@nagios:~# vi /usr/local/nagios/etc/services/localhost-swap.cfg
root@nagios:~# cat /usr/local/nagios/etc/services/localhost-swap.cfg
  define service{
    host_name                       localhost
    service_description             swap
    check_command                   check_swap
    check_interval                  5
    retry_interval                  1
    max_check_attempts              3
    check_period                    24x7
    contact_groups                  admins
    notification_interval           60
    notification_period             24x7
    notification_options            w,c,u,r
    }
```

```
root@nagios:~# vi /usr/local/nagios/etc/services/localhost-disk.cfg
root@nagios:~# cat /usr/local/nagios/etc/services/localhost-disk.cfg
  define service{
    host_name                       localhost
    service_description             disk
    check_command                   check_partition
    check_interval                  5
    retry_interval                  1
    max_check_attempts              3
    check_period                    24x7
```

```
    contact_groups                  admins
    notification_interval           60
    notification_period             24x7
    notification_options            w,c,u,r
    }
```

Vérifiez la configuration de Nagios :

```
root@nagios:~# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.2.0
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 08-01-2016
License: GPL

Website: https://www.nagios.org
Reading configuration data...
   Read main config file okay...
   Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
    Checked 9 services.
    Checked 2 hosts.
    Checked 0 host groups.
    Checked 0 service groups.
    Checked 1 contacts.
    Checked 2 contact groups.
    Checked 29 commands.
    Checked 3 time periods.
    Checked 0 host escalations.
    Checked 0 service escalations.
```

```
Checking for circular paths...
    Checked 2 hosts
    Checked 0 service dependencies
    Checked 0 host dependencies
    Checked 3 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors:   0

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

Re-démarrez le service nagios :

```
root@nagios:~# systemctl restart nagios
root@nagios:~# systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
   Loaded: loaded (/etc/init.d/nagios)
   Active: active (running) since Sun 2018-05-20 13:54:11 BST; 55s ago
  Process: 21852 ExecStop=/etc/init.d/nagios stop (code=exited, status=0/SUCCESS)
  Process: 21859 ExecStart=/etc/init.d/nagios start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/nagios.service
           ├─21879 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─21881 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─21882 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─21883 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─21884 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─21885 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

May 20 13:54:11 nagios.i2tch.loc nagios[21879]: nerd: Channel hostchecks registered successfully
May 20 13:54:11 nagios.i2tch.loc nagios[21879]: nerd: Channel servicechecks registered successfully
May 20 13:54:11 nagios.i2tch.loc nagios[21879]: nerd: Channel opathchecks registered successfully
```

```
May 20 13:54:11 nagios.i2tch.loc nagios[21879]: nerd: Fully initialized and ready to rock!
May 20 13:54:11 nagios.i2tch.loc nagios[21879]: wproc: Successfully registered manager as @wproc with query
handler
May 20 13:54:11 nagios.i2tch.loc nagios[21879]: wproc: Registry request: name=Core Worker 21883;pid=21883
May 20 13:54:11 nagios.i2tch.loc nagios[21879]: wproc: Registry request: name=Core Worker 21884;pid=21884
May 20 13:54:11 nagios.i2tch.loc nagios[21879]: wproc: Registry request: name=Core Worker 21882;pid=21882
May 20 13:54:11 nagios.i2tch.loc nagios[21879]: wproc: Registry request: name=Core Worker 21881;pid=21881
May 20 13:54:12 nagios.i2tch.loc nagios[21879]: Successfully launched command file worker with pid 21885
```

Lors de la vérification suivante par Nagios, l'état des services **swap** et **disk** devraient être **OK** :

```
root@nagios:~# ncmd -q list -h localhost
Fetching services and health on localhost
---
Service  State
---
disk     OK
ssh      OK
swap     OK
www      OK
```

La Charge Moyenne

Nagios est capable de surveiller la charge moyenne grâce au plugin **check_load**.

La syntaxe du plugin **check_load** est :

```
root@nagios:~# /usr/local/nagios/libexec/check_load --help
check_load v2.1.2 (nagios-plugins 2.1.2)
Copyright (c) 1999 Felipe Gustavo de Almeida <galmeida@linux.ime.usp.br>
Copyright (c) 1999-2014 Nagios Plugin Development Team
    <devel@nagios-plugins.org>
```

```
This plugin tests the current system load average.

Usage:
check_load [-r] -w WLOAD1,WLOAD5,WLOAD15 -c CLOAD1,CLOAD5,CLOAD15

Options:
 -h, --help
    Print detailed help screen
 -V, --version
    Print version information
 --extra-opts=[section][@file]
    Read options from an ini file. See
    https://www.nagios-plugins.org/doc/extra-opts.html
    for usage and examples.
 -w, --warning=WLOAD1,WLOAD5,WLOAD15
    Exit with WARNING status if load average exceeds WLOADn
 -c, --critical=CLOAD1,CLOAD5,CLOAD15
    Exit with CRITICAL status if load average exceed CLOADn
    the load average format is the same used by "uptime" and "w"
 -r, --percpu
    Divide the load averages by the number of CPUs (when possible)

Send email to help@nagios-plugins.org if you have questions regarding use
of this software. To submit patches or suggest improvements, send email to
devel@nagios-plugins.org
```

Créez le fichier de commande suivant :

```
root@nagios:~# vi /usr/local/nagios/etc/commands/load.cfg
root@nagios:~# cat /usr/local/nagios/etc/commands/load.cfg
 define command{
    command_name  check_load
    command_line  $USER1$/check_load -w 0.5,0.5,0.5 -c 0.7,0.7,0.7
```

```
  }
```

Créez ensuite le fichier **/usr/local/nagios/etc/services/localhost-load.cfg** :

```
root@nagios:~# vi /usr/local/nagios/etc/services/localhost-load.cfg
root@nagios:~# cat /usr/local/nagios/etc/services/localhost-load.cfg
  define service{
    host_name                       localhost
    service_description             load
    check_command                   check_load
    check_interval                  5
    retry_interval                  1
    max_check_attempts              3
    check_period                    24x7
    contact_groups                  admins
    notification_interval           60
    notification_period             24x7
    notification_options            w,c,u,r
    }
```

Vérifiez la configuration de Nagios :

```
root@nagios:~# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.2.0
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 08-01-2016
License: GPL

Website: https://www.nagios.org
Reading configuration data...
   Read main config file okay...
   Read object config files okay...
```

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

Checking objects...
    Checked 10 services.
    Checked 2 hosts.
    Checked 0 host groups.
    Checked 0 service groups.
    Checked 1 contacts.
    Checked 2 contact groups.
    Checked 30 commands.
    Checked 3 time periods.
    Checked 0 host escalations.
    Checked 0 service escalations.
Checking for circular paths...
    Checked 2 hosts
    Checked 0 service dependencies
    Checked 0 host dependencies
    Checked 3 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors:   0

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

Re-démarrez le service nagios :

```
root@nagios:~# systemctl restart nagios
root@nagios:~# systemctl status nagios
● nagios.service - LSB: Starts and stops the Nagios monitoring server
   Loaded: loaded (/etc/init.d/nagios)
   Active: active (running) since Sun 2018-05-20 14:32:28 BST; 6s ago
```

```
  Process: 28454 ExecStop=/etc/init.d/nagios stop (code=exited, status=0/SUCCESS)
  Process: 28461 ExecStart=/etc/init.d/nagios start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/nagios.service
           ├─28481 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           ├─28483 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─28484 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─28485 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           ├─28486 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─28487 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg

May 20 14:32:28 nagios.i2tch.loc nagios[28481]: nerd: Channel hostchecks registered successfully
May 20 14:32:28 nagios.i2tch.loc nagios[28481]: nerd: Channel servicechecks registered successfully
May 20 14:32:28 nagios.i2tch.loc nagios[28481]: nerd: Channel opathchecks registered successfully
May 20 14:32:28 nagios.i2tch.loc nagios[28481]: nerd: Fully initialized and ready to rock!
May 20 14:32:28 nagios.i2tch.loc nagios[28481]: wproc: Successfully registered manager as @wproc with query
handler
May 20 14:32:28 nagios.i2tch.loc nagios[28481]: wproc: Registry request: name=Core Worker 28486;pid=28486
May 20 14:32:28 nagios.i2tch.loc nagios[28481]: wproc: Registry request: name=Core Worker 28485;pid=28485
May 20 14:32:28 nagios.i2tch.loc nagios[28481]: wproc: Registry request: name=Core Worker 28484;pid=28484
May 20 14:32:28 nagios.i2tch.loc nagios[28481]: wproc: Registry request: name=Core Worker 28483;pid=28483
May 20 14:32:28 nagios.i2tch.loc nagios[28481]: Successfully launched command file worker with pid 28487
```

Lors de la vérification suivante par Nagios, l'état de la vérification **load** devrait être **OK** :

```
root@nagios:~# ncmd -q list -h localhost
Fetching services and health on localhost
---
Service  State
---
disk     OK
load     OK
ssh      OK
swap     OK
```

```
www       OK
```

Les Processus

Nagios est capable de surveiller les processus grâce au plugin **check_procs**.

La syntaxe du plugin **check_procs** est :

```
root@nagios:~# /usr/local/nagios/libexec/check_procs --help
check_procs v2.1.2 (nagios-plugins 2.1.2)
Copyright (c) 1999 Ethan Galstad <nagios@nagios.org>
Copyright (c) 2000-2014 Nagios Plugin Development Team
    <devel@nagios-plugins.org>

Checks all processes and generates WARNING or CRITICAL states if the specified
metric is outside the required threshold ranges. The metric defaults to number
of processes.  Search filters can be applied to limit the processes to check.


Usage:
check_procs -w <range> -c <range> [-m metric] [-s state] [-p ppid]
 [-u user] [-r rss] [-z vsz] [-P %cpu] [-a argument-array]
 [-C command] [-k] [-t timeout] [-v]

Options:
 -h, --help
    Print detailed help screen
 -V, --version
    Print version information
 --extra-opts=[section][@file]
    Read options from an ini file. See
    https://www.nagios-plugins.org/doc/extra-opts.html
    for usage and examples.
```

```
 -w, --warning=RANGE
   Generate warning state if metric is outside this range
 -c, --critical=RANGE
   Generate critical state if metric is outside this range
 -m, --metric=TYPE
  Check thresholds against metric. Valid types:
  PROCS   - number of processes (default)
  VSZ     - virtual memory size
  RSS     - resident set memory size
  CPU     - percentage CPU
  ELAPSED - time elapsed in seconds
 -t, --timeout=INTEGER
    Seconds before plugin times out (default: 10)
 -v, --verbose
    Extra information. Up to 3 verbosity levels
 -T, --traditional
   Filter own process the traditional way by PID instead of /proc/pid/exe

Filters:
 -s, --state=STATUSFLAGS
   Only scan for processes that have, in the output of `ps`, one or
   more of the status flags you specify (for example R, Z, S, RS,
   RSZDT, plus others based on the output of your 'ps' command).
 -p, --ppid=PPID
   Only scan for children of the parent process ID indicated.
 -z, --vsz=VSZ
   Only scan for processes with VSZ higher than indicated.
 -r, --rss=RSS
   Only scan for processes with RSS higher than indicated.
 -P, --pcpu=PCPU
   Only scan for processes with PCPU higher than indicated.
 -u, --user=USER
   Only scan for processes with user name or ID indicated.
 -a, --argument-array=STRING
```

```
   Only scan for processes with args that contain STRING.
 --ereg-argument-array=STRING
   Only scan for processes with args that contain the regex STRING.
 -C, --command=COMMAND
   Only scan for exact matches of COMMAND (without path).
 -k, --no-kthreads
   Only scan for non kernel threads (works on Linux only).
 -g, --cgroup-hierarchy
   Only scan for processes belonging to STRING hierarchy (works on Linux only).

RANGEs are specified 'min:max' or 'min:' or ':max' (or 'max'). If
specified 'max:min', a warning status will be generated if the
count is inside the specified range

This plugin checks the number of currently running processes and
generates WARNING or CRITICAL states if the process count is outside
the specified threshold ranges. The process count can be filtered by
process owner, parent process PID, current state (e.g., 'Z'), or may
be the total number of running processes

Examples:
 check_procs -w 2:2 -c 2:1024 -C portsentry
  Warning if not two processes with command name portsentry.
  Critical if < 2 or > 1024 processes

 check_procs -c 1:1 -C bind -g /
  Critical if not one processes with command name bind belonging to root cgroup.

 check_procs -w 10 -a '/usr/local/bin/perl' -u root
  Warning alert if > 10 processes with command arguments containing
  '/usr/local/bin/perl' and owned by root

 check_procs -w 50000 -c 100000 --metric=VSZ
  Alert if VSZ of any processes over 50K or 100K
```

```
 check_procs -w 10 -c 20 --metric=CPU
  Alert if CPU of any processes over 10%% or 20%%

Send email to help@nagios-plugins.org if you have questions regarding use
of this software. To submit patches or suggest improvements, send email to
devel@nagios-plugins.org
```

Créez le fichier de commande suivant :

```
root@nagios:~# vi /usr/local/nagios/etc/commands/procs.cfg
root@nagios:~# cat /usr/local/nagios/etc/commands/procs.cfg
define command{
    command_name  check_procs_cpu
    command_line  $USER1$/check_procs -w 10 -c 20 --metric=CPU
  }
```

Créez ensuite le fichier **/usr/local/nagios/etc/services/localhost-procs.cfg** :

```
root@nagios:~# vi /usr/local/nagios/etc/services/localhost-procs.cfg
root@nagios:~# cat /usr/local/nagios/etc/services/localhost-procs.cfg
  define service{
    host_name                       localhost
    service_description             procs
    check_command                   check_procs_cpu
    check_interval                  5
    retry_interval                  1
    max_check_attempts              3
    check_period                    24x7
    contact_groups                  admins
    notification_interval           60
    notification_period             24x7
    notification_options            w,c,u,r
    }
```

Vérifiez la configuration de Nagios :

```
root@nagios:~# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios Core 4.2.0
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 08-01-2016
License: GPL

Website: https://www.nagios.org
Reading configuration data...
   Read main config file okay...
   Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
    Checked 11 services.
    Checked 2 hosts.
    Checked 0 host groups.
    Checked 0 service groups.
    Checked 1 contacts.
    Checked 2 contact groups.
    Checked 31 commands.
    Checked 3 time periods.
    Checked 0 host escalations.
    Checked 0 service escalations.
Checking for circular paths...
    Checked 2 hosts
    Checked 0 service dependencies
    Checked 0 host dependencies
    Checked 3 timeperiods
Checking global event handlers...
```

```
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors:   0

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

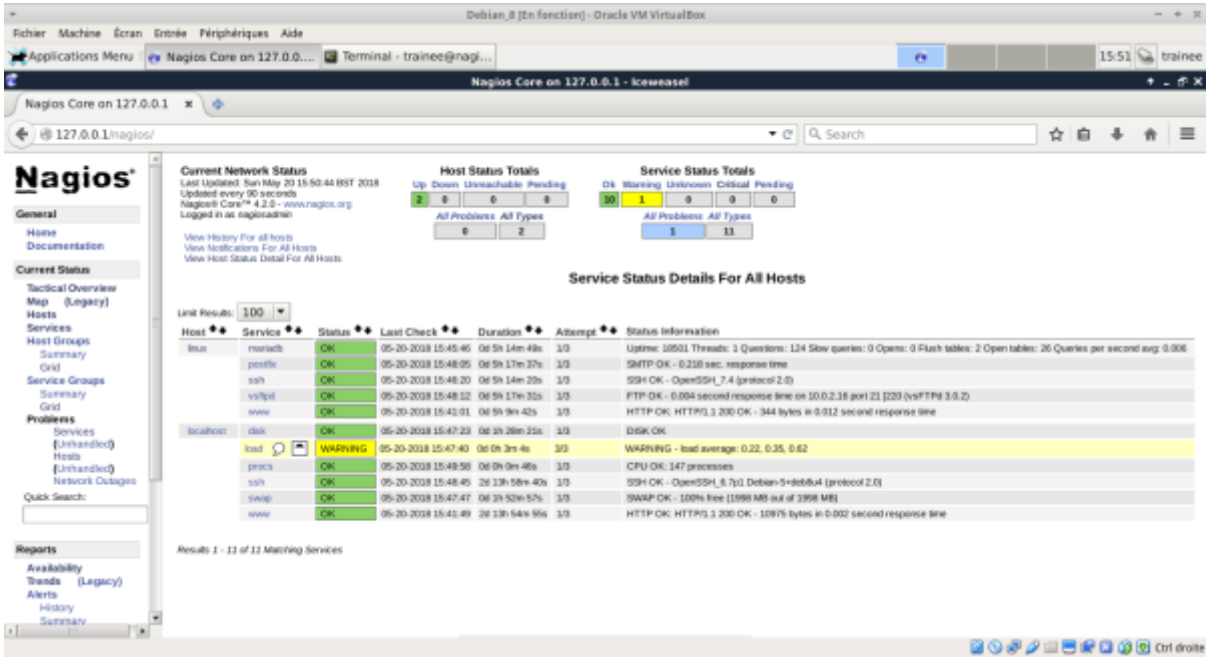
Re-démarrez le service nagios :

```
root@nagios:~# systemctl restart nagios
```

Lors de la vérification suivante par Nagios, l'état de la vérification **load** devrait être **OK** :

```
root@nagios:~# ncmd -q list -h localhost
Fetching services and health on localhost
---
Service  State
---
disk     OK
load     OK
procs    OK
ssh      OK
swap     OK
www      OK
```

Dernièrement, contrôlez l'état des vérifications en utilisant l'interface HTML. Vous devez obtenir un résultat similaire à celui ci-dessous :



SNMP

```
root@nagios:~# vi /etc/apt/sources.list
root@nagios:~# cat /etc/apt/sources.list
#

# deb cdrom:[Debian GNU/Linux 8.4.0 _Jessie_ - Official amd64 xfce-CD Binary-1 20160402-14:44]/ jessie main

#deb cdrom:[Debian GNU/Linux 8.4.0 _Jessie_ - Official amd64 xfce-CD Binary-1 20160402-14:44]/ jessie main

deb http://ftp.fr.debian.org/debian/ jessie main contrib non-free
deb-src http://ftp.fr.debian.org/debian/ jessie main contrib non-free

deb http://security.debian.org/ jessie/updates main contrib non-free
deb-src http://security.debian.org/ jessie/updates main contrib non-free
```

```
# jessie-updates, previously known as 'volatile'
deb http://ftp.fr.debian.org/debian/ jessie-updates main contrib non-free
deb-src http://ftp.fr.debian.org/debian/ jessie-updates main contrib non-free
```

```
root@nagios:~# apt-get update
root@nagios:~# apt-get autoremove
root@nagios:~# apt-get install snmpd snmp-mibs-downloader
```

```
root@nagios:~# snmpget -V
NET-SNMP version: 5.7.2.1
```

```
root@nagios:~# vi /etc/default/snmpd
root@nagios:~# cat /etc/default/snmpd
# This file controls the activity of snmpd

# Don't load any MIBs by default.
# You might comment this lines once you have the MIBs downloaded.
# export MIBS=/usr/share/mibs
export MIBDIRS=/usr/share/mibs

# snmpd control (yes means start daemon).
SNMPDRUN=yes

# snmpd options (use syslog, close stdin/out/err).
SNMPDOPTS='-Lsd -Lf /dev/null -u snmp -g snmp -I -smux,mteTrigger,mteTriggerConf -p /run/snmpd.pid'
```

```
root@nagios:~# snmpwalk -v 1 -c public 127.0.0.1
iso.3.6.1.2.1.1.1.0 = STRING: "Linux nagios.i2tch.loc 3.16.0-4-amd64 #1 SMP Debian 3.16.51-3 (2017-12-13) x86_64"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.8072.3.2.10
iso.3.6.1.2.1.1.3.0 = Timeticks: (4977) 0:00:49.77
iso.3.6.1.2.1.1.4.0 = STRING: "Me <me@example.org>"
iso.3.6.1.2.1.1.5.0 = STRING: "nagios.i2tch.loc"
iso.3.6.1.2.1.1.6.0 = STRING: "Sitting on the Dock of the Bay"
iso.3.6.1.2.1.1.7.0 = INTEGER: 72
```

```
iso.3.6.1.2.1.1.8.0 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.2.1 = OID: iso.3.6.1.6.3.11.3.1.1
iso.3.6.1.2.1.1.9.1.2.2 = OID: iso.3.6.1.6.3.15.2.1.1
iso.3.6.1.2.1.1.9.1.2.3 = OID: iso.3.6.1.6.3.10.3.1.1
iso.3.6.1.2.1.1.9.1.2.4 = OID: iso.3.6.1.6.3.1
iso.3.6.1.2.1.1.9.1.2.5 = OID: iso.3.6.1.2.1.49
iso.3.6.1.2.1.1.9.1.2.6 = OID: iso.3.6.1.2.1.4
iso.3.6.1.2.1.1.9.1.2.7 = OID: iso.3.6.1.2.1.50
iso.3.6.1.2.1.1.9.1.2.8 = OID: iso.3.6.1.6.3.16.2.2.1
iso.3.6.1.2.1.1.9.1.2.9 = OID: iso.3.6.1.6.3.13.3.1.3
iso.3.6.1.2.1.1.9.1.2.10 = OID: iso.3.6.1.2.1.92
iso.3.6.1.2.1.1.9.1.3.1 = STRING: "The MIB for Message Processing and Dispatching."
iso.3.6.1.2.1.1.9.1.3.2 = STRING: "The management information definitions for the SNMP User-based Security
Model."
iso.3.6.1.2.1.1.9.1.3.3 = STRING: "The SNMP Management Architecture MIB."
iso.3.6.1.2.1.1.9.1.3.4 = STRING: "The MIB module for SNMPv2 entities"
iso.3.6.1.2.1.1.9.1.3.5 = STRING: "The MIB module for managing TCP implementations"
iso.3.6.1.2.1.1.9.1.3.6 = STRING: "The MIB module for managing IP and ICMP implementations"
iso.3.6.1.2.1.1.9.1.3.7 = STRING: "The MIB module for managing UDP implementations"
iso.3.6.1.2.1.1.9.1.3.8 = STRING: "View-based Access Control Model for SNMP."
iso.3.6.1.2.1.1.9.1.3.9 = STRING: "The MIB modules for managing SNMP Notification, plus filtering."
iso.3.6.1.2.1.1.9.1.3.10 = STRING: "The MIB module for logging SNMP Notifications."
iso.3.6.1.2.1.1.9.1.4.1 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.2 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.3 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.4 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.5 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.6 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.7 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.8 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.9 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.10 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.25.1.1.0 = Timeticks: (46573) 0:07:45.73
iso.3.6.1.2.1.25.1.2.0 = Hex-STRING: 07 E3 02 06 0D 2A 0E 00 2B 00 00
```

```
iso.3.6.1.2.1.25.1.3.0 = INTEGER: 393216
iso.3.6.1.2.1.25.1.4.0 = STRING: "BOOT_IMAGE=/boot/vmlinuz-3.16.0-4-amd64 root=UUID=4a230056-285f-42f4-
bfe0-5a73dbc5b745 ro quiet
"
iso.3.6.1.2.1.25.1.5.0 = Gauge32: 2
iso.3.6.1.2.1.25.1.6.0 = Gauge32: 100
iso.3.6.1.2.1.25.1.7.0 = INTEGER: 0
End of MIB
```

```
[root@linux ~]# yum install net-snmp
```

```
[root@linux ~]# snmpget -V
NET-SNMP version: 5.7.2
```

```
[root@linux ~]# systemctl enable snmpd
```

```
[root@linux ~]# vi /etc/snmp/snmpd.conf
[root@linux ~]# cat /etc/snmp/snmpd.conf
...
com2sec local       localhost       rocommunity
com2sec mynetwork   10.0.2.0/24     rocommunity
...
```

```
[root@linux ~]# systemctl start snmpd
```

```
root@nagios:~# snmpwalk -v 1 -c public 10.0.2.16
iso.3.6.1.2.1.1.1.0 = STRING: "Linux linux.i2tch.loc 3.10.0-957.1.3.el7.x86_64 #1 SMP Thu Nov 29 14:49:43 UTC
2018 x86_64"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.8072.3.2.10
iso.3.6.1.2.1.1.3.0 = Timeticks: (1744) 0:00:17.44
iso.3.6.1.2.1.1.4.0 = STRING: "Root <root@localhost> (configure /etc/snmp/snmp.local.conf)"
iso.3.6.1.2.1.1.5.0 = STRING: "linux.i2tch.loc"
iso.3.6.1.2.1.1.6.0 = STRING: "Unknown (edit /etc/snmp/snmpd.conf)"
iso.3.6.1.2.1.1.8.0 = Timeticks: (45) 0:00:00.45
```

```
iso.3.6.1.2.1.1.9.1.2.1 = OID: iso.3.6.1.6.3.11.3.1.1
iso.3.6.1.2.1.1.9.1.2.2 = OID: iso.3.6.1.6.3.15.2.1.1
iso.3.6.1.2.1.1.9.1.2.3 = OID: iso.3.6.1.6.3.10.3.1.1
iso.3.6.1.2.1.1.9.1.2.4 = OID: iso.3.6.1.6.3.1
iso.3.6.1.2.1.1.9.1.2.5 = OID: iso.3.6.1.2.1.49
iso.3.6.1.2.1.1.9.1.2.6 = OID: iso.3.6.1.2.1.4
iso.3.6.1.2.1.1.9.1.2.7 = OID: iso.3.6.1.2.1.50
iso.3.6.1.2.1.1.9.1.2.8 = OID: iso.3.6.1.6.3.16.2.2.1
iso.3.6.1.2.1.1.9.1.2.9 = OID: iso.3.6.1.6.3.13.3.1.3
iso.3.6.1.2.1.1.9.1.2.10 = OID: iso.3.6.1.2.1.92
iso.3.6.1.2.1.1.9.1.3.1 = STRING: "The MIB for Message Processing and Dispatching."
iso.3.6.1.2.1.1.9.1.3.2 = STRING: "The management information definitions for the SNMP User-based Security
Model."
iso.3.6.1.2.1.1.9.1.3.3 = STRING: "The SNMP Management Architecture MIB."
iso.3.6.1.2.1.1.9.1.3.4 = STRING: "The MIB module for SNMPv2 entities"
iso.3.6.1.2.1.1.9.1.3.5 = STRING: "The MIB module for managing TCP implementations"
iso.3.6.1.2.1.1.9.1.3.6 = STRING: "The MIB module for managing IP and ICMP implementations"
iso.3.6.1.2.1.1.9.1.3.7 = STRING: "The MIB module for managing UDP implementations"
iso.3.6.1.2.1.1.9.1.3.8 = STRING: "View-based Access Control Model for SNMP."
iso.3.6.1.2.1.1.9.1.3.9 = STRING: "The MIB modules for managing SNMP Notification, plus filtering."
iso.3.6.1.2.1.1.9.1.3.10 = STRING: "The MIB module for logging SNMP Notifications."
iso.3.6.1.2.1.1.9.1.4.1 = Timeticks: (44) 0:00:00.44
iso.3.6.1.2.1.1.9.1.4.2 = Timeticks: (44) 0:00:00.44
iso.3.6.1.2.1.1.9.1.4.3 = Timeticks: (44) 0:00:00.44
iso.3.6.1.2.1.1.9.1.4.4 = Timeticks: (44) 0:00:00.44
iso.3.6.1.2.1.1.9.1.4.5 = Timeticks: (44) 0:00:00.44
iso.3.6.1.2.1.1.9.1.4.6 = Timeticks: (44) 0:00:00.44
iso.3.6.1.2.1.1.9.1.4.7 = Timeticks: (44) 0:00:00.44
iso.3.6.1.2.1.1.9.1.4.8 = Timeticks: (44) 0:00:00.44
iso.3.6.1.2.1.1.9.1.4.9 = Timeticks: (45) 0:00:00.45
iso.3.6.1.2.1.1.9.1.4.10 = Timeticks: (45) 0:00:00.45
iso.3.6.1.2.1.25.1.1.0 = Timeticks: (809727) 2:14:57.27
End of MIB
```

```
root@nagios:~# snmpget -v 1 -c public 10.0.2.16 1.3.6.1.2.1.1.1.0
iso.3.6.1.2.1.1.1.0 = STRING: "Linux linux.i2tch.loc 3.10.0-957.1.3.el7.x86_64 #1 SMP Thu Nov 29 14:49:43 UTC
2018 x86_64"
```

```
root@nagios:~# snmpget -v 2c -c public 10.0.2.16 1.3.6.1.2.1.1.1.0
iso.3.6.1.2.1.1.1.0 = STRING: "Linux linux.i2tch.loc 3.10.0-957.1.3.el7.x86_64 #1 SMP Thu Nov 29 14:49:43 UTC
2018 x86_64"
```

```
root@nagios:~# snmpget -v 3 -c public 10.0.2.16 1.3.6.1.2.1.1.1.0
snmpget: No securityName specified
```

```
root@nagios:~# /usr/local/nagios/libexec/check_snmp -H 10.0.2.16 -P 2c -o 1.3.6.1.2.1.1.1.0
SNMP OK - "Linux linux.i2tch.loc 3.10.0-957.1.3.el7.x86_64 #1 SMP Thu Nov 29 14:49:43 UTC 2018 x86_64" |
```
