PARTIE DHCP SERVER

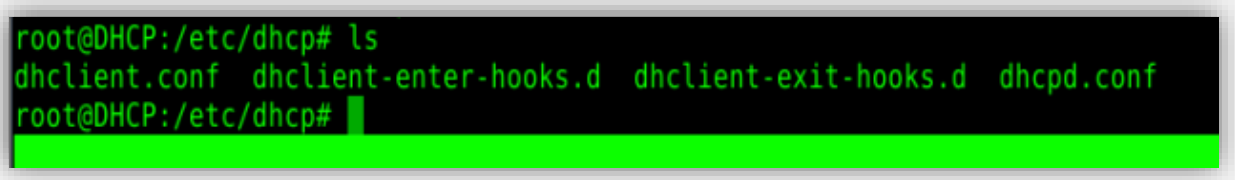
Faire les prérequis d'installation :

Apt-get update & upgrade

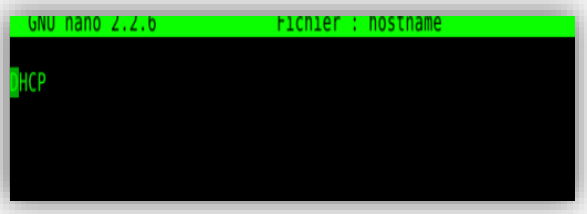
Installer les paquets du serveur dhcp

apt-get install isc-dhcp-server

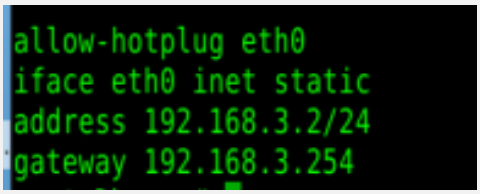
Après l'installation, les fichiers ci-dessous doivent apparaitre :



Changer le nom de la machine dans ***/etc/hostname***



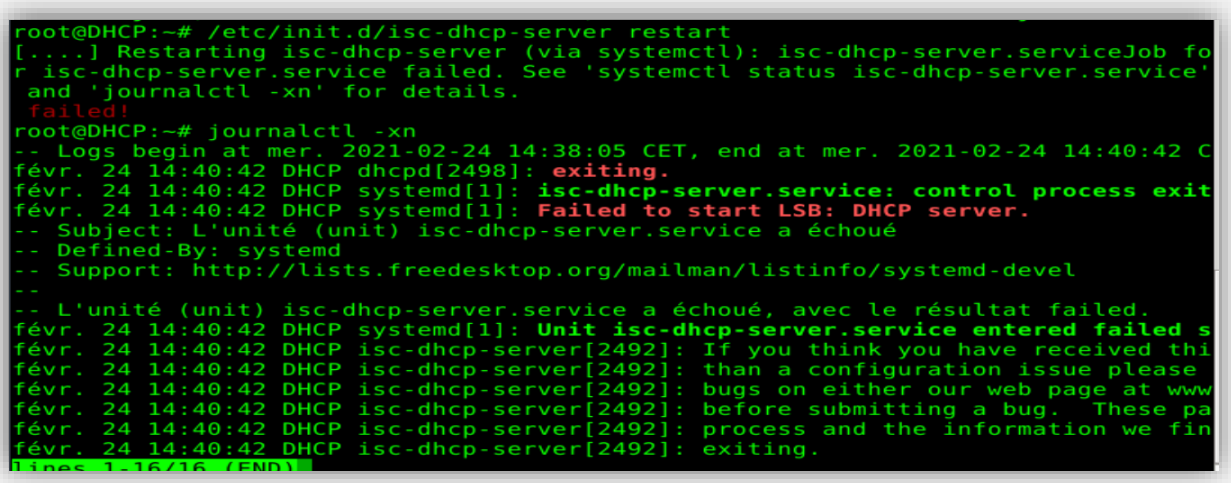
Modifier l'interface dans le fichier qui se trouve dans ***/etc/network/interfaces***



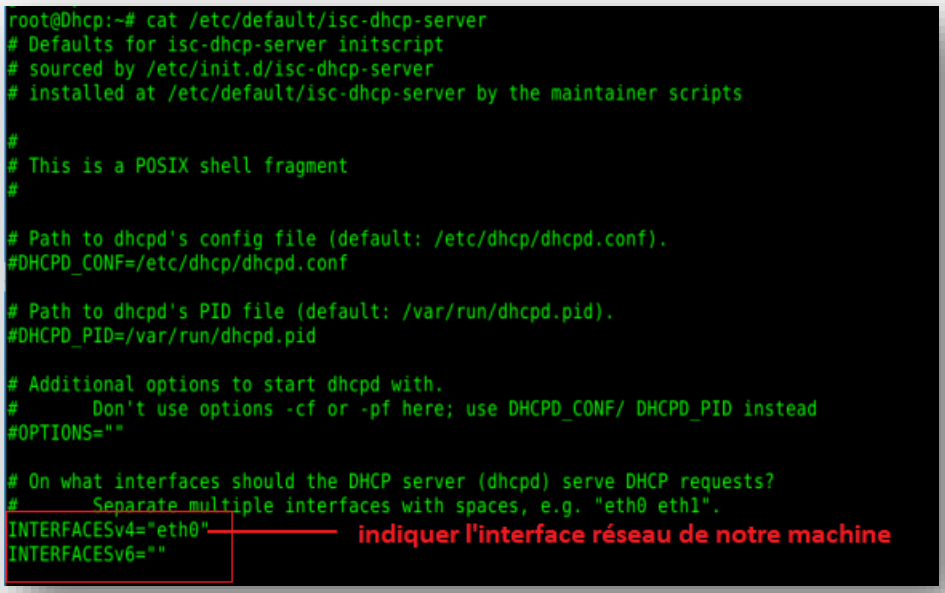
Faire un ***reboot*** pour relancer la machine !

Relancer le serveur DHCP :

Vous allez apercevoir ce message d'erreur :

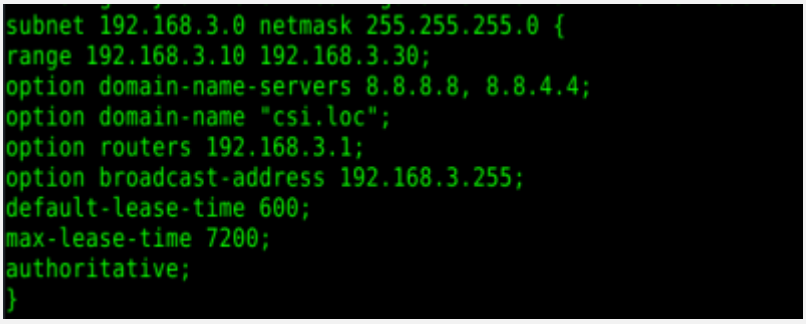


Se rendre dans le chemin ci-dessous et modifier le fichier ***isc-dhcp-server*** :

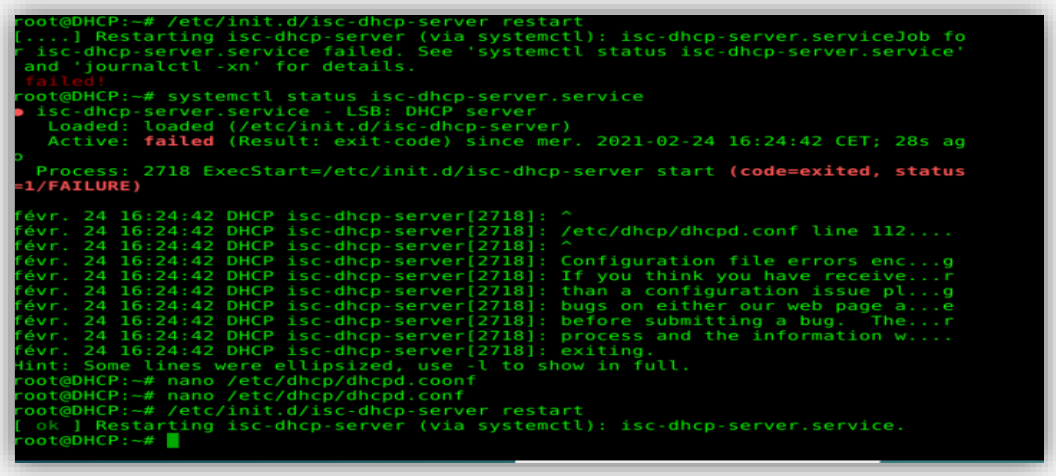


Dans le fichier de paramétrage du dhcp

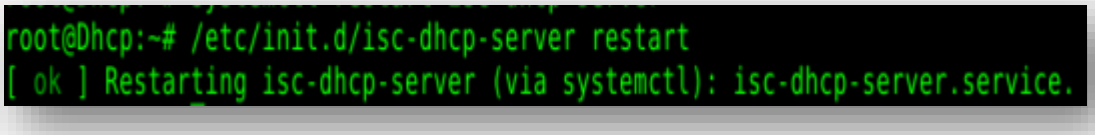
Décommenter ces lignes et remplacer en fonction de la plage Ip qu'on souhaite attribuer :



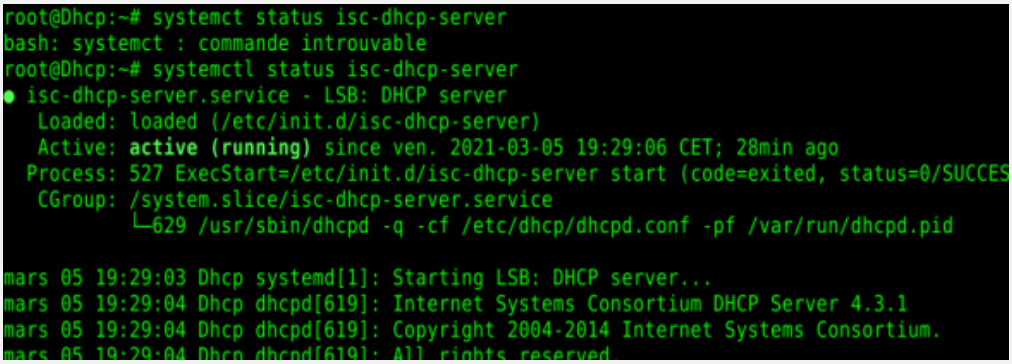
Attention ne pas négliger les caractères spéciaux, en cas de problème exécuter la commande ci-dessous pour voir d'où vient le problème :



Ensuite relancer le service après avoir rectifié la ligne de notre fichier

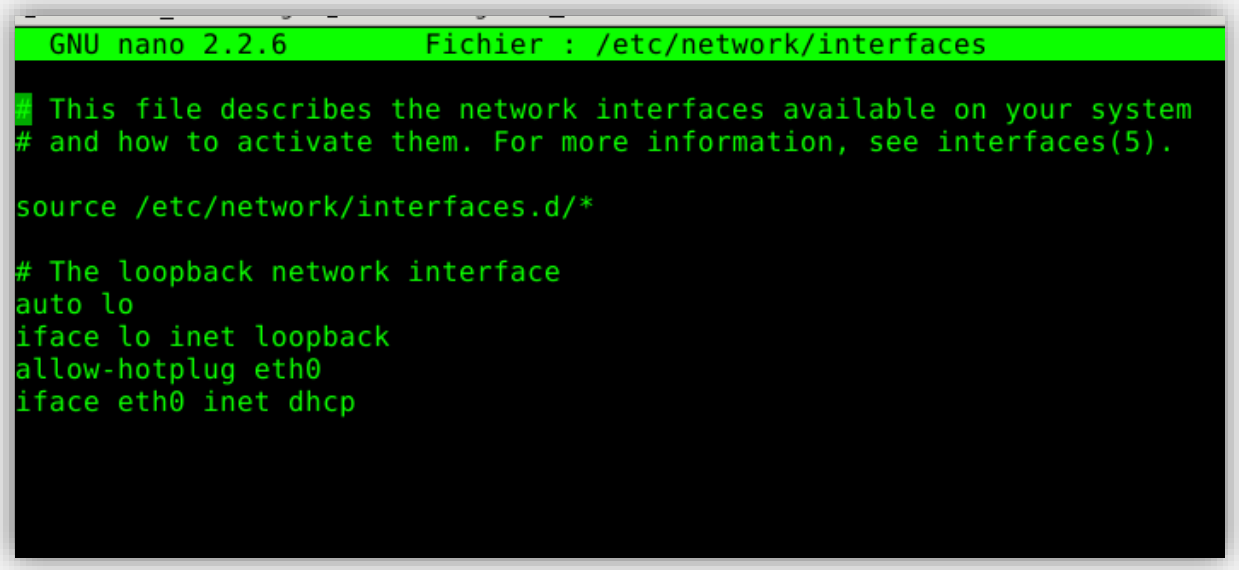


Maintenant on peut voir que tout se passe bien après l'avoir relancé :

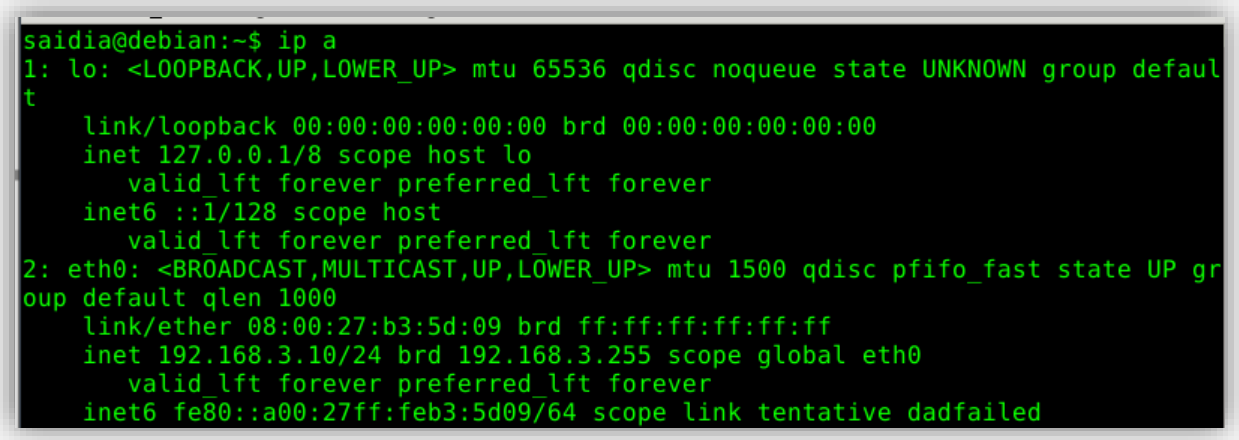


Test sur machine Client :

Configurer l'interface de cette façon l'interface du client :



Relancer la machine et vérifier qu'une Ip a bien été attribuée :



Parfait tout fonctionne à merveille !!! 😊

PARTIE RELAIS DHCP

Installer et indiquer l'adresse IP du serveur DHCP précédent :

```
root@debian:~# apt-get install isc-dhcp-relay
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les NOUVEAUX paquets suivants seront installés :
  isc-dhcp-relay
0 mis à jour, 1 nouvellement installés, 0 à enlever et 1 n
Il est nécessaire de prendre 198 ko dans les archives.
Après cette opération, 512 ko d'espace disque supplémentai
ATTENTION : les paquets suivants n'ont pas été authentifié
  isc-dhcp-relay
Faut-il installer ces paquets sans vérification ? [o/N]
```

Se rendre dans le fichier interfaces :

```
 GNU nano 2.2.6        Fichier : /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto eth0
iface eth0 inet loopback
address 172.20.38.61
netmask 255.255.255.192


#2eme interface
auto eth1
iface eth1 inet static
address 172.20.38.126
netmask 255.255.255.192
```

Relancer l'interface internet :

```
root@RELAIS-DHCP:~# service networking restart
```

Décommenter cette commande dans le fichier ***/etc/sysctl.conf***

```
net.ipv4.ip_forward=1
```

Activer le routage

```
root@RELAIS-DHCP:~# sysctl -p /etc/sysctl.conf
net.ipv4.ip_forward = 1
net.ipv4.conf.all.log_martians = 1
```

Indiquer les interfaces de notre machine comme ci-dessous :

```
  GNU nano 2.2.6       Fichier : /etc/default/isc-dhcp-relay

# Defaults for isc-dhcp-relay initscript
# sourced by /etc/init.d/isc-dhcp-relay
# installed at /etc/default/isc-dhcp-relay by the maintainer scripts

#
# This is a POSIX shell fragment
#

# What servers should the DHCP relay forward requests to?
SERVERS="192.168.3.2"

# On what interfaces should the DHCP relay (dhrelay) serve DHCP requests?
INTERFACES="eth0 eth1""

# Additional options that are passed to the DHCP relay daemon?
OPTIONS=""
```

Puis relancer le service isc-dhcp-relay :

```
root@RELAIS-DHCP:~# service isc-dhcp-relay restart
root@RELAIS-DHCP:~#
```

S’assurer du fonctionnement :

```
root@RELAIS-DHCP:~# systemctl status isc-dhcp-relay
● isc-dhcp-relay.service - LSB: DHCP relay
   Loaded: loaded (/etc/init.d/isc-dhcp-relay)
   Active: active (running) since ven. 2021-03-05 02:15:01 CET; 3s ago
  Process: 3117 ExecStop=/etc/init.d/isc-dhcp-relay stop (code=exited, status=0/SUCCESS)
  Process: 3120 ExecStart=/etc/init.d/isc-dhcp-relay start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/isc-dhcp-relay.service
           └─3124 /usr/sbin/dhcrelay -q -i eth0 -i eth0 172.20.38.1

mars 05 02:15:01 RELAIS-DHCP systemd[1]: Starting LSB: DHCP relay...
mars 05 02:15:01 RELAIS-DHCP systemd[1]: Started LSB: DHCP relay.
root@RELAIS-DHCP:~# service isc-dhcp-relay restart
root@RELAIS-DHCP:~# service isc-dhcp-relay stop
root@RELAIS-DHCP:~# service isc-dhcp-relay start
root@RELAIS-DHCP:~# systemctl status isc-dhcp-relay
● isc-dhcp-relay.service - LSB: DHCP relay
   Loaded: loaded (/etc/init.d/isc-dhcp-relay)
   Active: active (running) since ven. 2021-03-05 02:16:12 CET; 4s ago
  Process: 3158 ExecStop=/etc/init.d/isc-dhcp-relay stop (code=exited, status=0/SUCCESS)
  Process: 3182 ExecStart=/etc/init.d/isc-dhcp-relay start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/isc-dhcp-relay.service
           └─3185 /usr/sbin/dhcrelay -q -i eth0 -i eth0 172.20.38.1

mars 05 02:16:12 RELAIS-DHCP systemd[1]: Started LSB: DHCP relay.
root@RELAIS-DHCP:~#
```

Se rendre sur le chemin ***nano /etc/sysctl.conf***

Décommenter cette ligne :

```
# Log martian packets
net.ipv4.conf.all.log_martians = 1
```

Activer le routage avec la commande ci-dessous :

```
root@RELAIS-DHCP:~# sysctl -p /etc/sysctl.conf
net.ipv4.conf.all.log_martians = 1
```

Supervision

Pour installer notre stratégie de supervision nous avons procédé de cette façon :

Prérequis :

apt-get install apache2 apt-get install build-essential unzip opensslibssl-dev libgd2-xpm-dev xinetd apache2-utils

Utilisateur :

useraddnagios groupaddnagcmd usermod -a -G

nagcmdnagios&&usermod -a -G nagcmd www-data

Pour la décompression du fichier nous utilisons

tar -xvf nagios-4.*.tar.gz

Puis nous rentrons dans le fichier cd nagios-4.2.2/et nous effectuons ces commandes :

./configure --with-nagios-group=nagios --with-command-group=nagcmd make all

make

install

make

install-init

make

install-

config

make install-

commandmode

a2enmod a2enmod cgi

Copie du virtualhost avec l'attribution des permissions et son activation:

cpsample-config/httpd.conf/etc/apache2/sites-available/nagios4.conf

chmod 644 /etc/apache2/sites-available/nagios.conf a2ensite nagios.conf

Création du fichier passwd pour nagios:

htpasswd -c /usr/local/nagios/etc/htpasswd.usersnagiosadmin

service apache2 start

Pour la décompression du fichier on installe des plugins :

tar -xvf nagios-plugins-2*.tar.gz

Pour la configuration et l'installation des Paquets:

cd nagios-plugins-2.*

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

openssl make

make install

Création du service nagiossystemd:

File: /etc/systemd/system/nagios.service

[Unit]

Description=Nagios

BindTo=network.target

[Install]

WantedBy=multi-user.target

[Service]

User=nagios

Group=nagios

Type=simple

ExecStart=/usr/local/nagios/bin/nagios /usr/local/nagios/etc/nagios.cfg

Pour finir on lance le service Nagios :

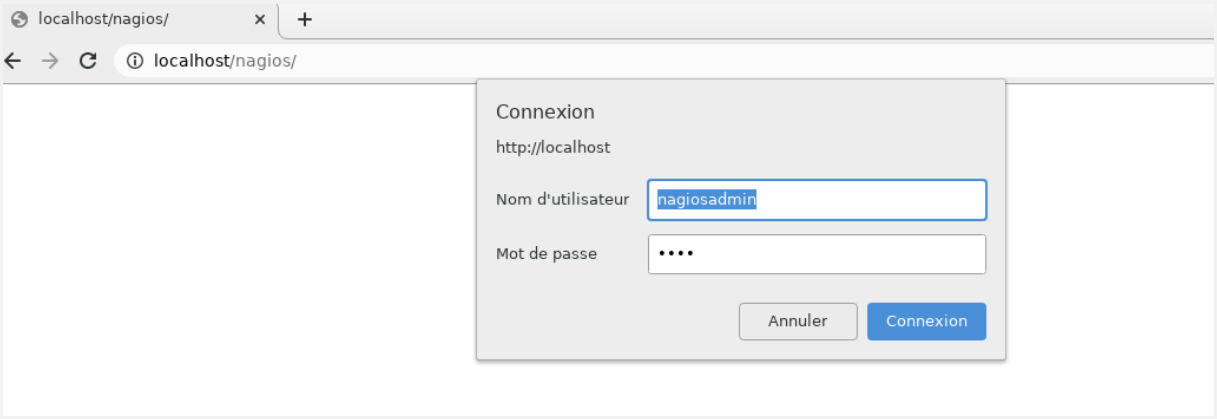
systemctl enable /etc/systemd/system/nagios.service systemctl start nagios

systemctl status nagios

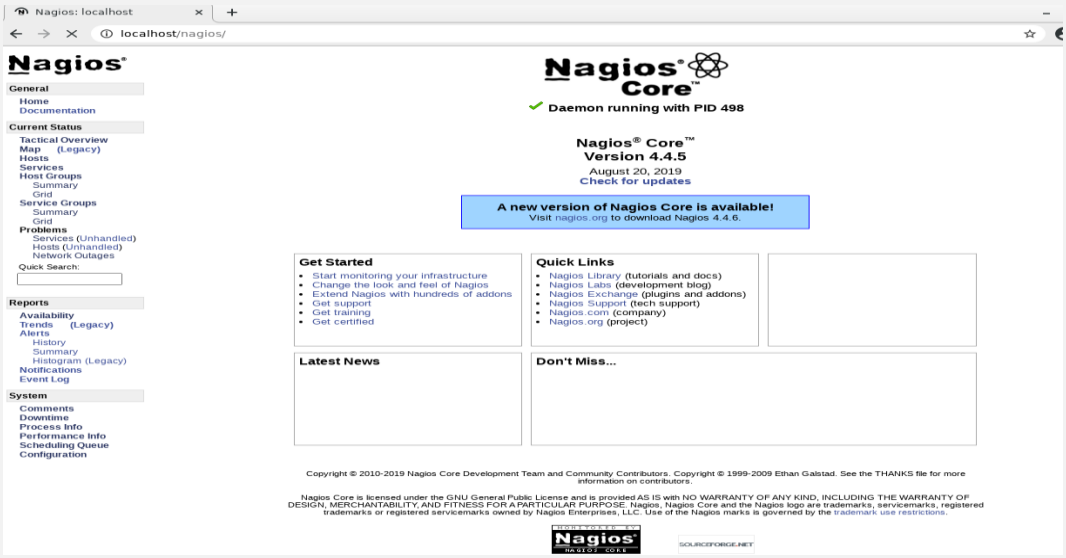
```
root@debnagios:~# service nagios status
● nagios.service - LSB: Starts and stops the Nagios monitoring server
   Loaded: loaded (/etc/init.d/nagios)
   Active: active (running) since jeu. 2017-12-07 07:45:36 CET; 2 days ago
  Process: 24210 ExecStop=/etc/init.d/nagios stop (code=exited, status=0/SUCCESS
```

On se connecte à l'interface graphique. Nous devons utiliser notre adresse ip suivie de /nagios/ ce qui donne : " localhost/nagios/" puis nous devons renseigner le login et le mot de passe qui sont :

- login = nagiosadmin
- mdp = nagios



On sera alors dirigés sur cette page :



Maintenant il faut créer des fichiers afin de superviser notre serveur DHCP.

Pour cela nous avons copié le fichier *localhost.cfg* par exemple en le nommant comme on le souhaite.

***/usr/local/nagios*/etc/objects/deb3.cfg**

```
##############################################################################

# Define the dhcp server deb that we'll be monitoring

define host {

    use                     linux-server                    ; Inher
    host_name               deb3                    ; The name we're
    alias                   localhost               ; A longer name as
    address                 192.168.3.2                     ; IP ad
}
```

On spécifie l'adresse du serveur et les différents services à superviser au choix :

```
# Define a service to "ping" the local machine

define service{
        use                             local-service         ; Name of service template to use
        host_name                       deb3
        service_description             PING
        check_command                   check_ping!100.0,20%!500.0,60%
        }


# Define a service to check the disk space of the root partition
# on the local machine.  Warning if < 20% free, critical if
# < 10% free space on partition.

define service{
        use                             local-service         ; Name of service template to use
        host_name                       deb3
        service_description             Root Partition
        check_command                   check_local_disk!20%!10%!/
        }



# Define a service to check the number of currently logged in
# users on the local machine.  Warning if > 20 users, critical
# if > 50 users.

define service{
        use                             local-service         ; Name of service template to use
```

```
define service{
        use                             local-service         ; Name of service template to use
        host_name                       deb3
        service_description             Total Processes
        check_command                   check_local_procs!250!400!RSZDT
        }



# Define a service to check the load on the local machine.

define service{
        use                             local-service         ; Name of service template to use
        host_name                       deb3
        service_description             Current Load
        check_command                   check_local_load!5.0,4.0,3.0!10.0,6.0,4.0
        }



# Define a service to check the swap usage the local machine.
# Critical if less than 10% of swap is free, warning if less than 20% is free

define service{
        use                             local-service         ; Name of service template to use
        host_name                       deb3
        service_description             Swap Usage
        check_command                   check_local_swap!20!10
```

Il ne faut pas oublier d'ajouter les services manquants dans ***commands.cfg***

```
define command{
        command_name    check_https
        command_line $USER1$/check_http -H $HOSTADDRESS$ -S -p 443

        }
```

-S étant SSL

-P étant le port, ici 443 pour HTTPS.

```
define command{
        command_name    check_dns
        command_line    $USER1$/check_dns -H www.msf3.fr -s $HOSTADDRESS$
        }
```

Ce fichier nous permet de checker les services suivants : FTP, PING, SSH, DNS, HTTPS, HTTP…

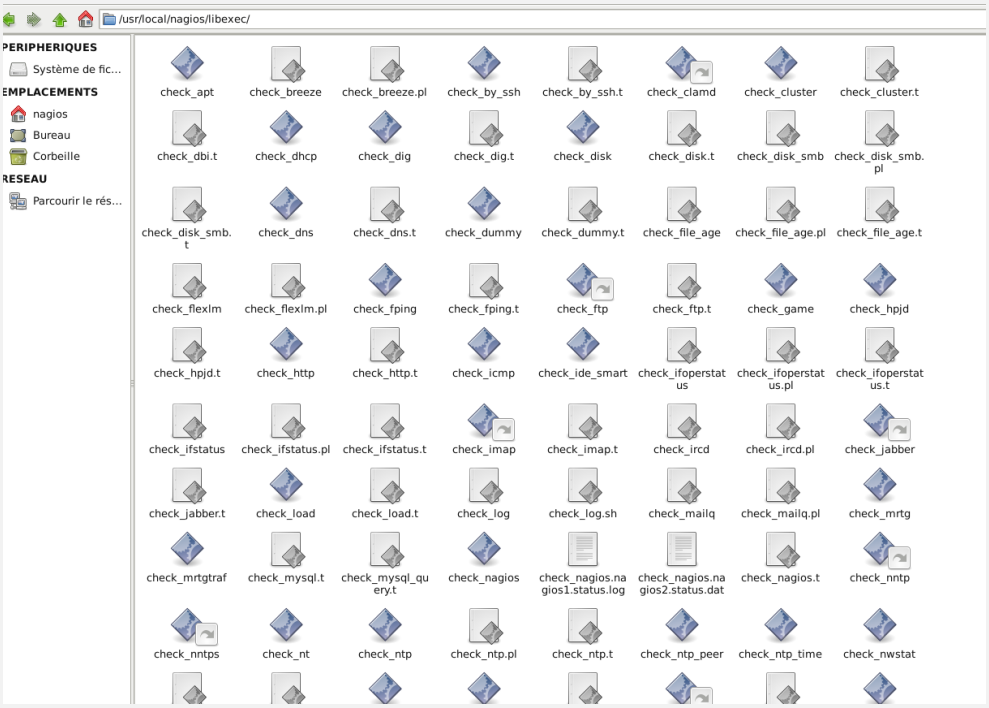
```
define host{
        use             windows-server  ; Inherit default values from a template
        host_name       winserver       ; The name we're giving to this host
        alias           My Windows Server       ; A longer name associated with the host
        address         172.18.158.140  ; IP address of the host
        }
```

```
define service{
        use                     generic-service
        host_name               winserver
        service_description     NSClient++ Version
        check_command           check_nt!CLIENTVERSION
}



# Create a service for monitoring the uptime of the server
# Change the host_name to match the name of the host you defined ab

define service{
        use                     generic-service
        host_name               winserver
        service_description     Uptime
        check_command           check_nt!UPTIME
        }



# Create a service for monitoring CPU load
# Change the host_name to match the name of the host you defined ab

define service{
        use                     generic-service
        host_name               winserver
        service_description     CPU Load
        check_command           check_nt!CPULOAD!-l 5,80,90
        }
```

```
define service{
        use                     generic-service
        host_name               ad
        service_description     Memory Usage
        check_command           check_nt!MEMUSE!-w 80 -c 90
        }


define service{
        use                     generic-service
        host_name               ad
        service_description     C:\ Drive Space
        check_command           check_nt!USEDDISKSPACE!-l c -w 80 -c 90
        }
define service{
        use                     generic-service
        host_name               ad
        service_description     Explorer
        check_command           check_nt!PROCSTATE!-d SHOWALL -l explorer.exe
        }
```

Ne pas oublier de télécharger les plugins et de les mettre dans le chemin ci-dessous

(Vue de l'interface graphique):



Pour activer ces deux fichiers il faut aller dans

/usr/local/nagios/etc/nagios.cfg et ajouter le chemin des deux fichiers que l'on a configurés.

```
  GNU nano 2.2.6               Fichier : nagios.cfg                    Modi

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

# Definitions for monitoring the local (Linux) host
cfg_file=/usr/local/nagios/etc/objects/localhost.cfg

# Definitions for monitoring a Windows machine
cfg_file=/usr/local/nagios/etc/objects/windows.cfg

# Definitions for monitoring a router/switch
cfg_file=/usr/local/nagios/etc/objects/switch.cfg
#cfg_file=/usr/local/nagios/etc/objects/router.cfg
cfg_file=/usr/local/nagios/etc/objects/deb3.cfg

# Definitions for monitoring a network printer
#cfg_file=/usr/local/nagios/etc/objects/printer.cfg


# You can also tell Nagios to process all config files (with a .cfg
```

Pour vérifier que nos fichiers sont bien configurés nous avons utilisé cette commande :

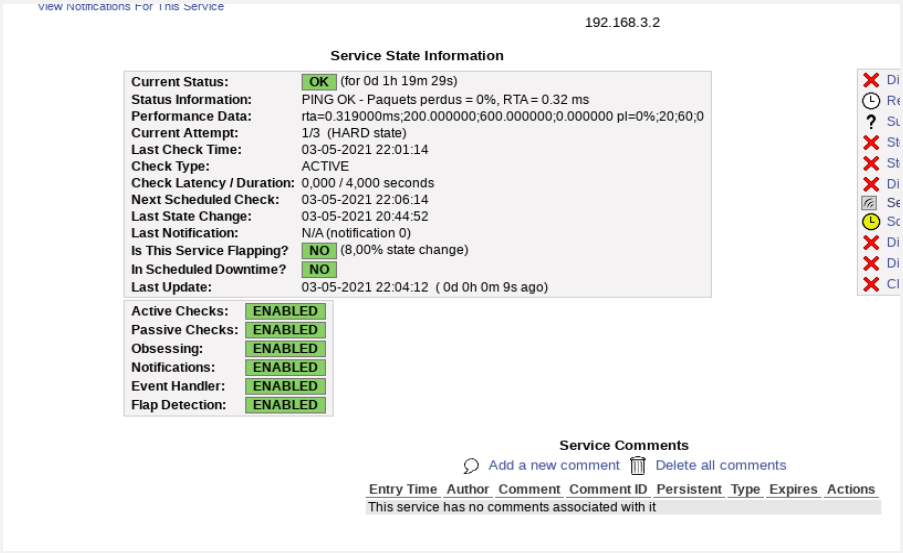
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

On remarque qu'il n'y a pas d'erreurs, ensuite relancer nagios :

```
service nagios restart
```

```
● nagios.service - Nagios Core 4.4.5
   Loaded: loaded (/lib/systemd/system/nagios.service; enabled)
   Active: active (running) since ven. 2021-03-05 22:01:42 CET; 1min 36s ago
     Docs: https://www.nagios.org/documentation
  Process: 3130 ExecStopPost=/bin/rm -f /usr/local/nagios/var/rw/nagios.cmd (
e=exited, status=0/SUCCESS)
  Process: 3128 ExecStop=/bin/kill -s TERM ${MAINPID} (code=exited, status=0/
CESS)
  Process: 3135 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/e
nagios.cfg (code=exited, status=0/SUCCESS)
  Process: 3133 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagio
tc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 3138 (nagios)
   CGroup: /system.slice/nagios.service
           ├─3138 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagio
           ├─3139 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var
```

Nous vérifions sur l'interface graphique si la machine est fonctionnelle :



Mission accomplie !!! 😉