# DOCUMENTATION CENTREON

# I. Installation NRPE sur Centos7

## 1. Add the EPEL repository

Lignes de commande :

```
> yum install epel-release
```

## 2. Install NRPE and the plugins that is required to add the services.

```
> yum install nrpe nagios-plugins-users nagios-plugins-load
> nagios-plugins-swap nagios-plugins-disk nagios-plugins-procs
```

## 3. Configure the agent to utilize the plugins.

Create a new file called **/etc/nrpe.d/op5_commands.cfg** containing the following information:

```
################################################################################
########
#
# op5-nrpe command configuration file
#
# COMMAND DEFINITIONS
# Syntax:
# command[<command_name>]=<command_line>
#
command[users]=/usr/lib64/nagios/plugins/check_users -w 5 -c 10
command[load]=/usr/lib64/nagios/plugins/check_load -w 15,10,5 -c
30,25,20
command[check_load]=/usr/lib64/nagios/plugins/check_load -w 15,10,5 -c
30,25,20
command[swap]=/usr/lib64/nagios/plugins/check_swap -w 20% -c 10%
command[root_disk]=/usr/lib64/nagios/plugins/check_disk -w 20% -c 10% -p
/ -m
command[usr_disk]=/usr/lib64/nagios/plugins/check_disk -w 20% -c 10% -p
/usr -m
command[var_disk]=/usr/lib64/nagios/plugins/check_disk -w 20% -c 10% -p
/var -m
command[zombie_procs]=/usr/lib64/nagios/plugins/check_procs -w 5 -c 10
-s Z
command[total_procs]=/usr/lib64/nagios/plugins/check_procs -w 190 -c 200
command[proc_named]=/usr/lib64/nagios/plugins/check_procs -w 1: -c 1:2
-C named
command[proc_crond]=/usr/lib64/nagios/plugins/check_procs -w 1: -c 1:5
-C crond
command[proc_syslogd]=/usr/lib64/nagios/plugins/check_procs -w 1: -c 1:2
-C syslog-ng
command[proc_rsyslogd]=/usr/lib64/nagios/plugins/check_procs -w 1: -c
1:2 -C rsyslogd
```

## 4. Now edit /etc/nagios/nrpe.cfg and change all text by :

```
################################################################################
#####
# Sample NRPE Config File
# Written by: Ethan Galstad (nagios@nagios.org)
#
# Last Modified: 2016-05-10
#
# NOTES:
# This is a sample configuration file for the NRPE daemon. It needs to
be
# located on the remote host that is running the NRPE daemon, not the
host
# from which the check_nrpe client is being executed.
################################################################################
#####
# LOG FACILITY
# The syslog facility that should be used for logging purposes.
log_facility=daemon
```

```
# LOG FILE
# If a log file is specified in this option, nrpe will write to
# that file instead of using syslog.
#log_file=/var/run/nrpe.log
# DEBUGGING OPTION
# This option determines whether or not debugging messages are logged to
the
# syslog facility.
# Values: 0=debugging off, 1=debugging on
debug=0
# PID FILE
# The name of the file in which the NRPE daemon should write it's
process ID
# number. The file is only written if the NRPE daemon is started by the
root
# user and is running in standalone mode.
pid_file=/var/run/nrpe/nrpe.pid
# PORT NUMBER
# Port number we should wait for connections on.
# NOTE: This must be a non-privileged port (i.e. > 1024).
# NOTE: This option is ignored if NRPE is running under either inetd or
xinetd
server_port=5666
# SERVER ADDRESS
# Address that nrpe should bind to in case there are more than one
interface
# and you do not want nrpe to bind on all interfaces.
# NOTE: This option is ignored if NRPE is running under either inetd or
xinetd
#server_address=127.0.0.1
# LISTEN QUEUE SIZE
# Listen queue size (backlog) for serving incoming connections.
# You may want to increase this value under high load.
#listen_queue_size=5
# NRPE USER
# This determines the effective user that the NRPE daemon should run as.
# You can either supply a username or a UID.
#
# NOTE: This option is ignored if NRPE is running under either inetd or
xinetd
nrpe_user=nrpe
# NRPE GROUP
# This determines the effective group that the NRPE daemon should run
as.
# You can either supply a group name or a GID.
#
# NOTE: This option is ignored if NRPE is running under either inetd or
xinetd
nrpe_group=nrpe
# 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
allowed_hosts=127.0.0.1
# COMMAND ARGUMENT PROCESSING
```

```
# This option determines whether or not the NRPE daemon will allow
clients
# to specify arguments to commands that are executed. This option only
works
# if the daemon was configured with the --enable-command-args configure
script
# option.
#
# *** ENABLING THIS OPTION IS A SECURITY RISK! ***
# Read the SECURITY file for information on some of the security
implications
# of enabling this variable.
#
# Values: 0=do not allow arguments, 1=allow command arguments
dont_blame_nrpe=0
# BASH COMMAND SUBSTITUTION
# This option determines whether or not the NRPE daemon will allow
clients
# to specify arguments that contain bash command substitutions of the
form
# $(...). This option only works if the daemon was configured with both
# the --enable-command-args and --enable-bash-command-substitution
configure
# script options.
#
# *** ENABLING THIS OPTION IS A HIGH SECURITY RISK! ***
# Read the SECURITY file for information on some of the security
implications
# of enabling this variable.
#
# Values: 0=do not allow bash command substitutions,
# 1=allow bash command substitutions
allow_bash_command_substitution=0
# COMMAND PREFIX
# This option allows you to prefix all commands with a user-defined
string.
# A space is automatically added between the specified prefix string and
the
# command line from the command definition.
#
# *** THIS EXAMPLE MAY POSE A POTENTIAL SECURITY RISK, SO USE WITH
CAUTION! ***
# Usage scenario:
# Execute restricted commmands using sudo. For this to work, you need
to add
# the nagios user to your /etc/sudoers. An example entry for allowing
# execution of the plugins from might be:
#
# nagios ALL=(ALL) NOPASSWD: /usr/lib/nagios/plugins/
#
# This lets the nagios user run all commands in that directory (and only
them)
# without asking for a password. If you do this, make sure you don't
give
# random users write access to that directory or its contents!
# command_prefix=/usr/bin/sudo
# COMMAND TIMEOUT
# This specifies the maximum number of seconds that the NRPE daemon will
# allow plugins to finish executing before killing them off.
command_timeout=60
# CONNECTION TIMEOUT
# This specifies the maximum number of seconds that the NRPE daemon will
# wait for a connection to be established before exiting. This is
sometimes
# seen where a network problem stops the SSL being established even though
# all network sessions are connected. This causes the nrpe daemons to
# accumulate, eating system resources. Do not set this too low.
```

```
connection_timeout=300
# WEAK RANDOM SEED OPTION
# This directive allows you to use SSL even if your system does not have
# a /dev/random or /dev/urandom (on purpose or because the necessary
patches
# were not applied). The random number generator will be seeded from a
file
# which is either a file pointed to by the environment valiable
$RANDFILE
# or $HOME/.rnd. If neither exists, the pseudo random number generator
will
# be initialized and a warning will be issued.
# Values: 0=only seed from /dev/[u]random, 1=also seed from weak
randomness
#allow_weak_random_seed=1
# SSL/TLS OPTIONS
# These directives allow you to specify how to use SSL/TLS.
# SSL VERSION
# This can be any of: SSLv2 (only use SSLv2), SSLv2+ (use any version),
# SSLv3 (only use SSLv3), SSLv3+ (use SSLv3 or above), TLSv1
(only use
# TLSv1), TLSv1+ (use TLSv1 or above), TLSv1.1 (only use
TLSv1.1),
# TLSv1.1+ (use TLSv1.1 or above), TLSv1.2 (only use TLSv1.2),
# TLSv1.2+ (use TLSv1.2 or above)
# If an "or above" version is used, the best will be negotiated. So if
both
# ends are able to do TLSv1.2 and use specify SSLv2, you will get
TLSv1.2.
# If you are using openssl 1.1.0 or above, the SSLv2 options are not
available.
#ssl_version=SSLv2+
# SSL USE ADH
# This is for backward compatibility and is DEPRECATED. Set to 1 to
enable
# ADH or 2 to require ADH. 1 is currently the default but will be
changed
# in a later version.
#ssl_use_adh=1
# SSL CIPHER LIST
# This lists which ciphers can be used. For backward compatibility, this
# defaults to 'ssl_cipher_list=ALL:!MD5:@STRENGTH' in this version but
# will be changed to something like the example below in a later version
of NRPE.
#ssl_cipher_list=ALL:!MD5:@STRENGTH
#ssl_cipher_list=ALL:!aNULL:!eNULL:!SSLv2:!LOW:!EXP:!RC4:!MD5:@STRENGTH
# SSL Certificate and Private Key Files
#ssl_cacert_file=/etc/ssl/servercerts/ca-cert.pem
#ssl_cert_file=/etc/ssl/servercerts/nagios-cert.pem
#ssl_privatekey_file=/etc/ssl/servercerts/nagios-key.pem
# SSL USE CLIENT CERTS
# This options determines client certificate usage.
# Values: 0 = Don't ask for or require client certificates (default)
# 1 = Ask for client certificates
# 2 = Require client certificates
#ssl_client_certs=0
# SSL LOGGING
# This option determines which SSL messages are send to syslog. OR
values
# together to specify multiple options.
# Values: 0x00 (0) = No additional logging (default)
# 0x01 (1) = Log startup SSL/TLS parameters
# 0x02 (2) = Log remote IP address
# 0x04 (4) = Log SSL/TLS version of connections
# 0x08 (8) = Log which cipher is being used for the connection
# 0x10 (16) = Log if client has a certificate
# 0x20 (32) = Log details of client's certificate if it has one
```

```
# -1 or 0xff or 0x2f = All of the above
#ssl_logging=0x00
# NASTY METACHARACTERS
# This option allows you to override the list of characters that cannot
# be passed to the NRPE daemon.
# nasty_metachars="|`&><'\\[]{};\r\n"
# INCLUDE CONFIG FILE
# This directive allows you to include definitions from an external
config file.
#include=<somefile.cfg>
# INCLUDE CONFIG DIRECTORY
# This directive allows you to include definitions from config files
(with a
# .cfg extension) in one or more directories (with recursion).
include_dir=/etc/nrpe.d/
# COMMAND DEFINITIONS
# Command definitions that this daemon will run. Definitions
# are in the following format:
#
# command[<command_name>]=<command_line>
#
# When the daemon receives a request to return the results of
<command_name>
# it will execute the command specified by the <command_line> argument.
#
# Unlike Nagios, the command line cannot contain macros - it must be
# typed exactly as it should be executed.
#
# Note: Any plugins that are used in the command lines must reside
# on the machine that this daemon is running on! The examples below
# assume that you have plugins installed in a /usr/local/nagios/libexec
# directory. Also note that you will have to modify the definitions
below
# to match the argument format the plugins expect. Remember, these are
# examples only!
# The following examples use hardcoded command arguments...
command[check_users]=/usr/lib64/nagios/plugins/check_users -w 5 -c 10
command[check_load]=/usr/lib64/nagios/plugins/check_load -r -w
.15,.10,.05 -c .30,.25,.20
command[check_hda1]=/usr/lib64/nagios/plugins/check_disk -w 20% -c 10%
-p /dev/hda1
command[check_zombie_procs]=/usr/lib64/nagios/plugins/check_procs -w 5
-c 10 -s Z
command[check_total_procs]=/usr/lib64/nagios/plugins/check_procs -w 150
-c 200
command[check_wildfly_procs]=/usr/lib64/nagios/plugins/check_procs -w 3:
-c 1: -u wildfly
command[check_java_procs]=/usr/lib64/nagios/plugins/check_procs -c 1: -C
java
command[check_mysql_procs]=/usr/lib64/nagios/plugins/check_procs -c 1:
-u mysql
command[check_apache_procs]=/usr/lib64/nagios/plugins/check_procs -w 5:
-c 1: -u apache
command[check_wildfly_http]=/usr/lib64/nagios/plugins/check_http -I
localhost -p 8080
command[check_sso_client_login_http]=/usr/lib64/nagios/plugins/check_htt
p -H 172.16.15.122 -p 9001 -u /client/login -T "application/json" -j
POST -P '{"clientId":"uaa-echosens-client",
"clientSecret":"EMPrhXTUxpcM"}' -v
# The following examples allow user-supplied arguments and can
# only be used if the NRPE daemon was compiled with support for
# command arguments *AND* the dont_blame_nrpe directive in this
# config file is set to '1'. This poses a potential security risk, so
# make sure you read the SECURITY file before doing this.
#command[check_users]=/usr/lib64/nagios/plugins/check_users -w $ARG1$ -c
$ARG2$
#command[check_load]=/usr/lib64/nagios/plugins/check_load -w $ARG1$ -c
```

```
$ARG2$
#command[check_disk]=/usr/lib64/nagios/plugins/check_disk -w $ARG1$ -c
$ARG2$ -p $ARG3$
#command[check_procs]=/usr/lib64/nagios/plugins/check_procs -w $ARG1$ -c
$ARG2$ -s $ARG3
```

5. Now edit line 106 and add your Monitor server(s) address(es) to the allowed_hosts parameter as a comma-separated list, example:

```
> allowed_hosts=127.0.0.1,10.0.0.10,10.0.0.11
```

6. Restart the nrpe agent on the host, and make sure that nrpe is started at boot:

```
> systemctl restart nrpe
> systemctl enable nrpe0
```

7. Install nagios plugins

*a. install common prerequisites*

```
> yum install -y gcc glibc glibc-common make gettext automake autoconf wget openssl-devel net-
 snmp net-snmp-utils epel-release
> yum install -y perl-Net-SNMP
```

*b. Downloading the source*

```
> cd /tmp
> wget --no-check-certificate -O nagios-plugins.tar.gz
> https://github.com/nagios-plugins/nagios-plugins/archive/release-2.2.1.tar.gz
> tar zxf nagios-plugins.tar.gz
```

*c. Compile and install*

```
> cd /tmp/nagios-plugins-release-2.2.1/
./tools/setup
./configure
> make
> make install
```

*d. Copy all files from "/usr/local/nagios/libexec" to "/usr/lib64/nagios/plugins/" (remplace existing files)*

*e. Change permissions for execute files*

```
> cd /usr/lib64/nagios/plugins
> chmod 755 *
```

# II. Installation de Centreon

Edit the file "etc/selinux/config" and remplace "enforcing" by "disabled"

```
> SELINUX= disabled
```

Reboot the server

```
> reboot
```

Make a update

```
> yum update
```

Install "wget"

```
> yum install wget
```

Do Centreon install commands

```
> wget http://yum.centreon.com/standard/3.4/el7/stable/noarch/RPMS/centreon-release-
  3.44.el7.centos.noarch.rpm
> yum install --nogpgcheck
> centreon-release-3.4-4.el7.centos.noarch.rpm
> yum install centreon-base-config-centreon-engine centreon
```

Install Maria BD

```
> yum install MariaDB-server
```

Restart mySql service

```
> service mysqld restart
```

Create a file "php-timezone.ini" in "/etc/php.d" and add this line

```
> date.timezone = Europe/Paris
```

Create a folder "mariadb.service.d" in "/etc/systemd/system". And in this folder create a file named "limits.conf" and edit with "[Service]\nLimitNOFILE=32000\n"

```
> mkdir -p /etc/systemd/system/mariadb.service.d
> echo -ne "[Service]\nLimitNOFILE=32000\n" | tee
  /etc/systemd/system/mariadb.service.d/limits.conf
```

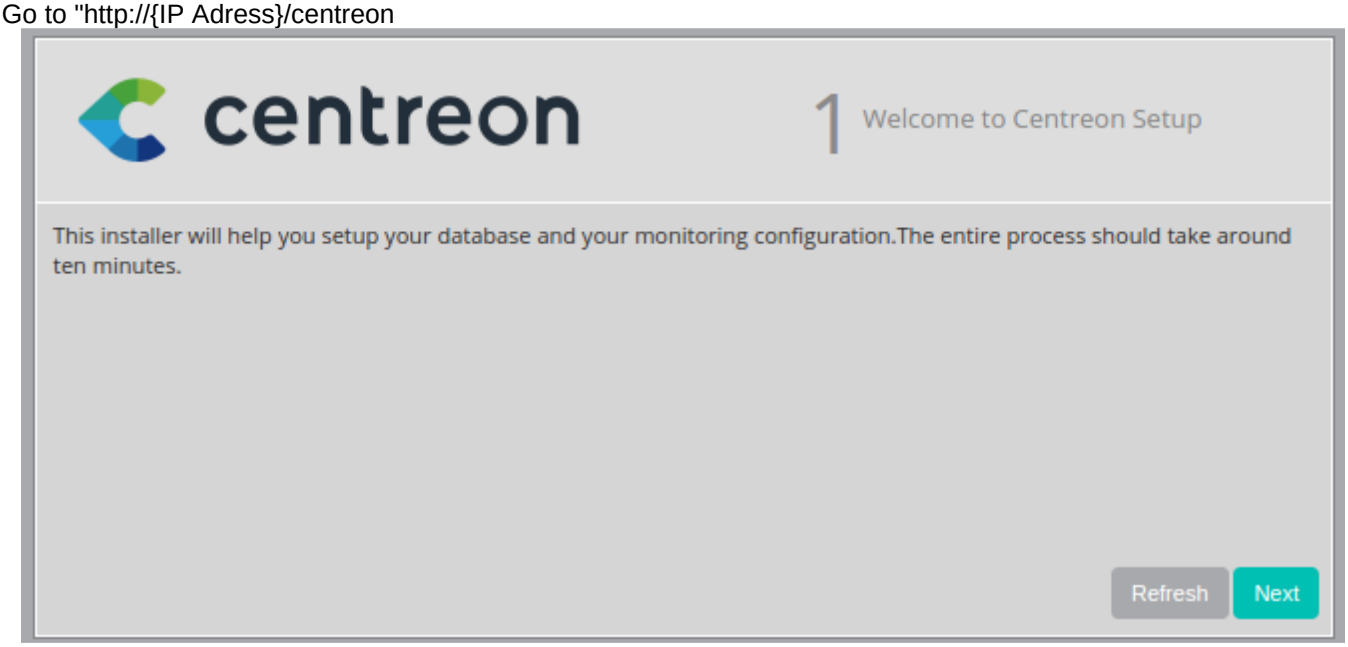
Reload services

```
> systemctl daemon-reload
> service mysqld restart
> service httpd restart
```

Stop & disable firewalld

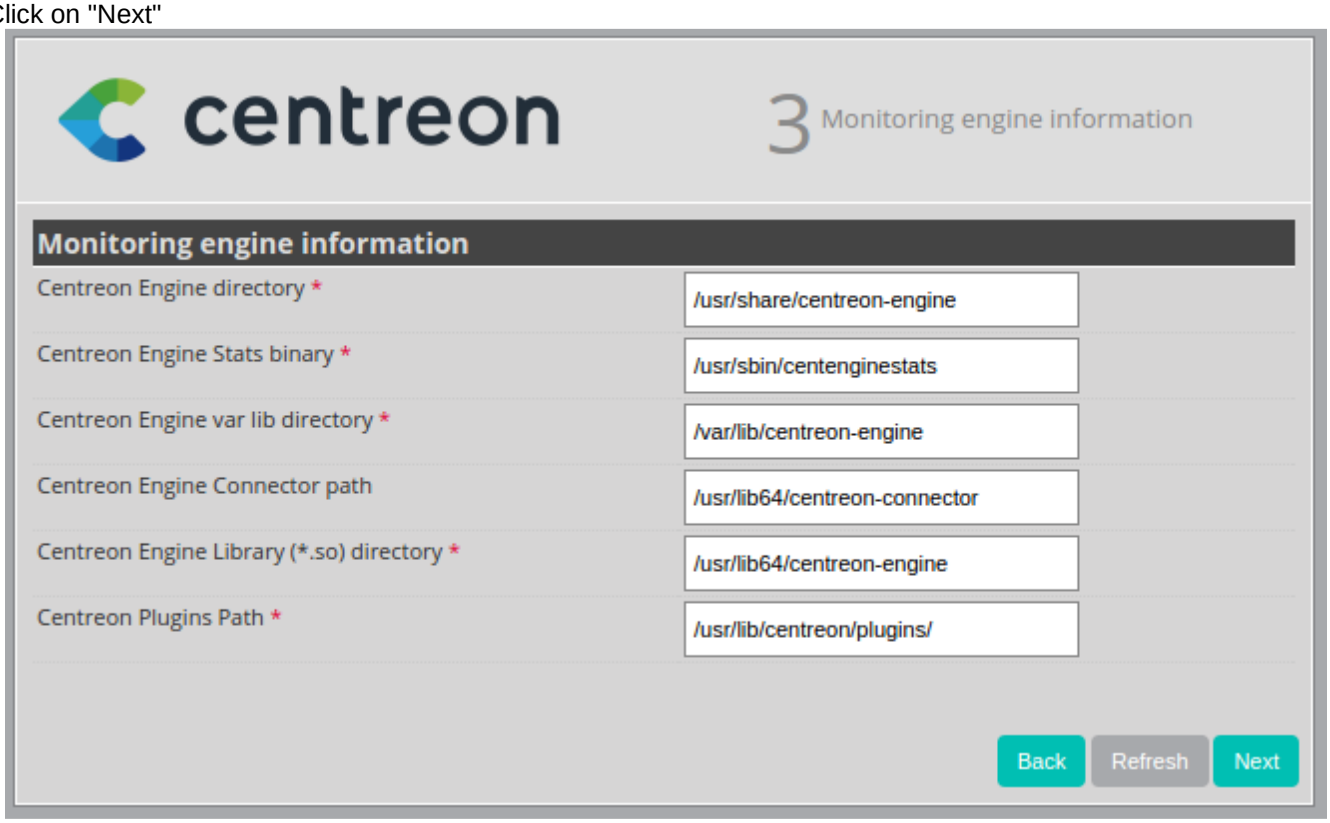
```
> service firewalld stop
> service firewalld disable
```

Go to "http://{IP Adress}/centreon

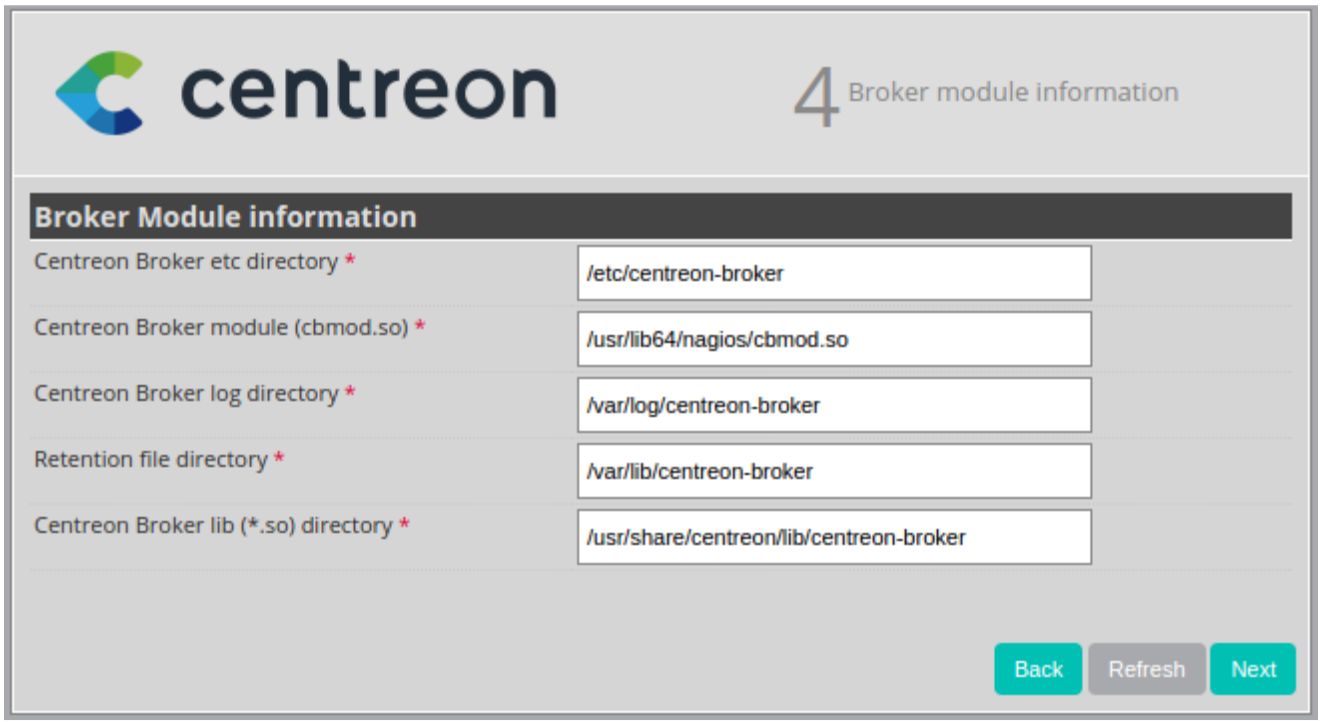


Click on "Next"

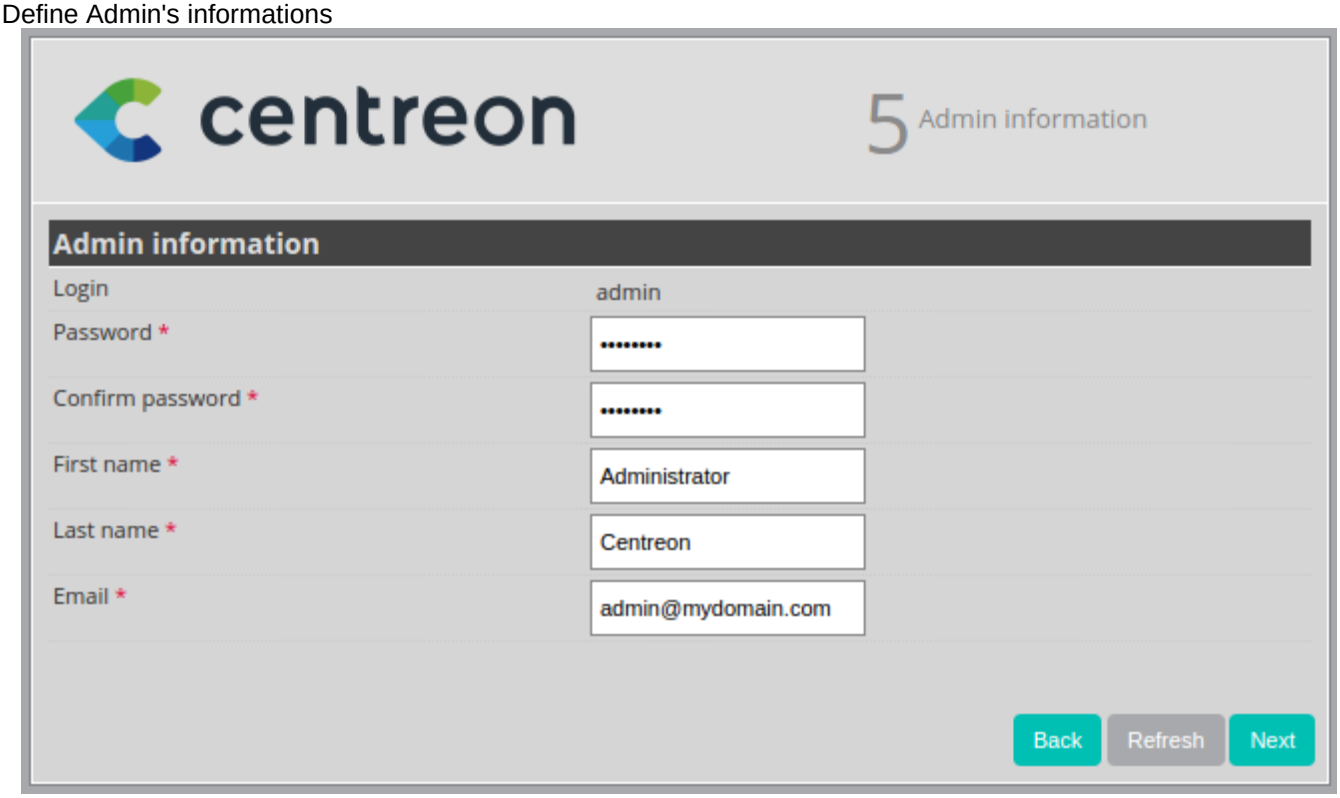

Click on "Next"



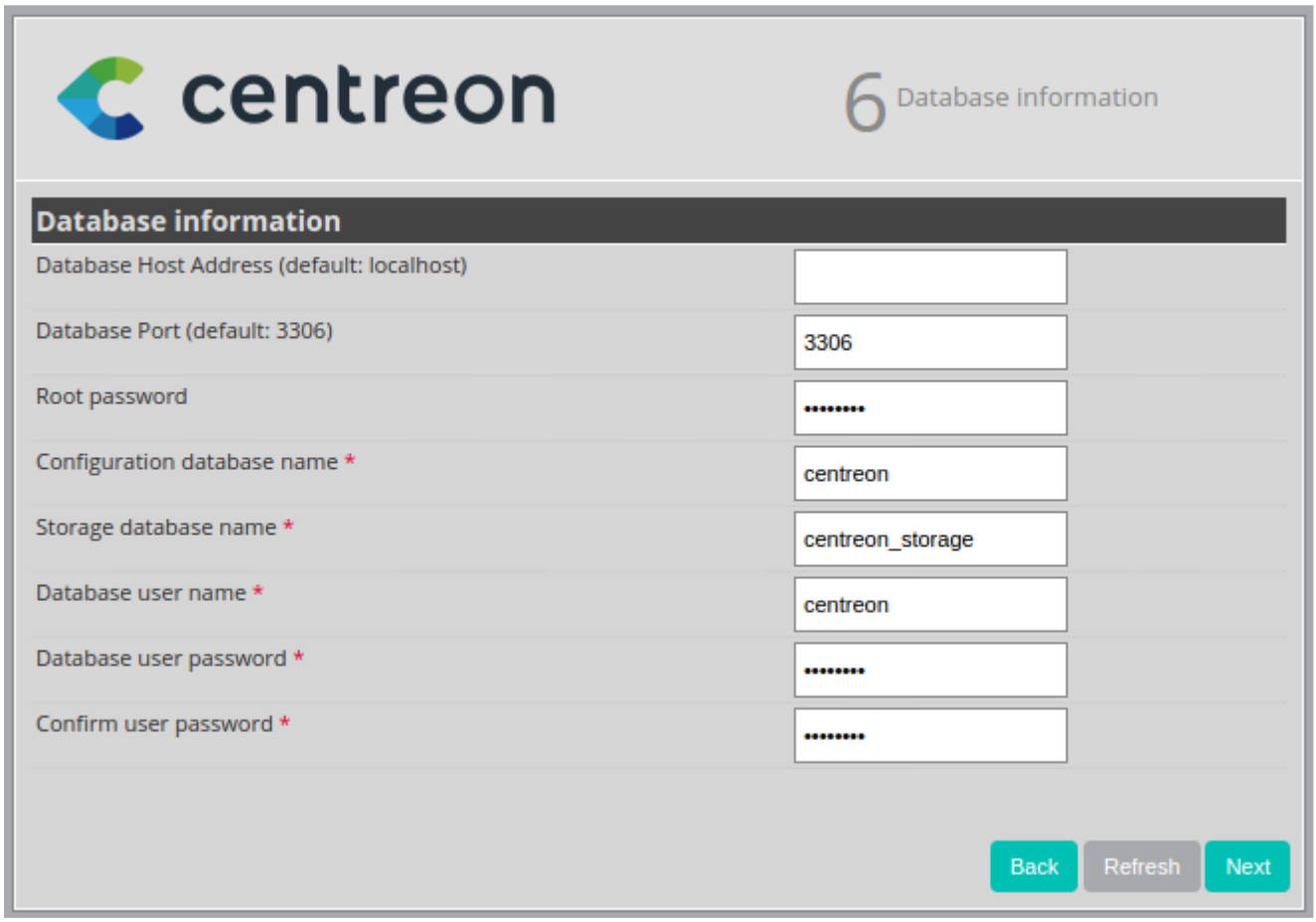

Click on "Next"



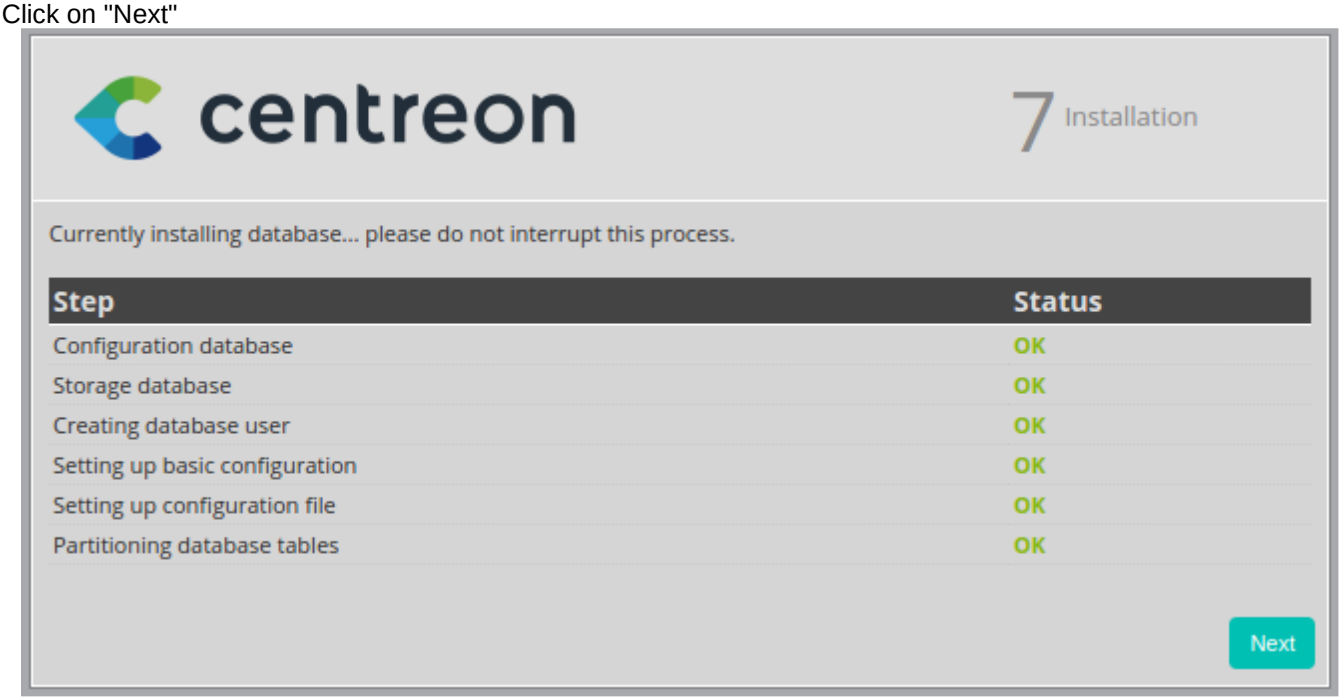

Define Admin's informations




Define a database user password




Click on "Next"




Click on "Finish"


Now, you can log in

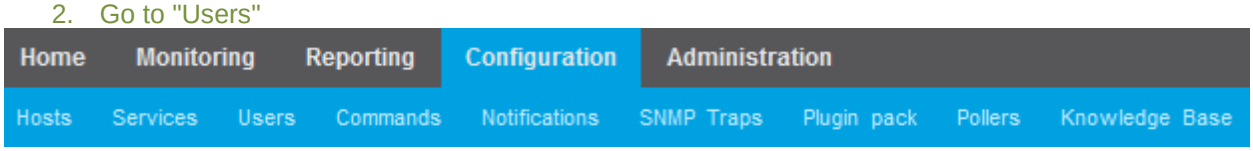

Install Centreon NRPE plugin

```
> yum install centreon-nrpe-plugin
```

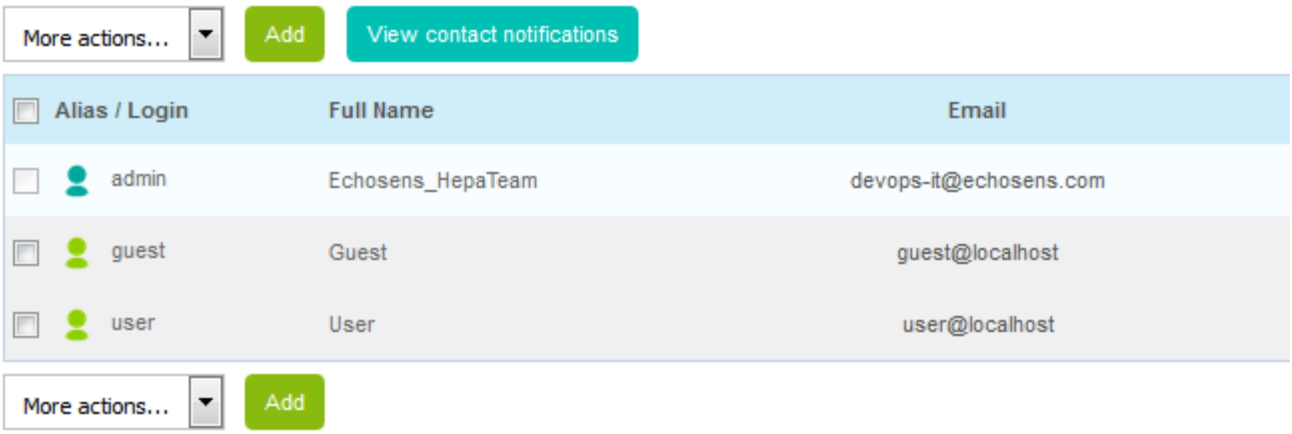
Update User

1. In Centreon, go to "Configuration

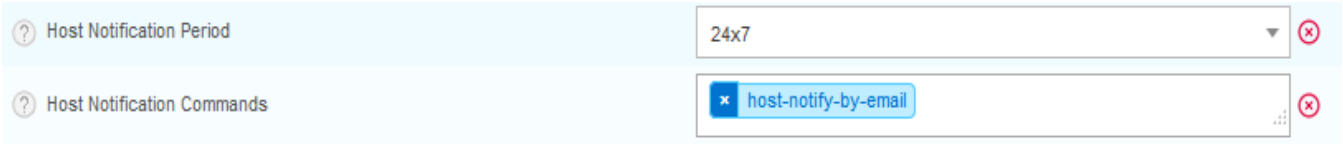

2. Go to "Users"



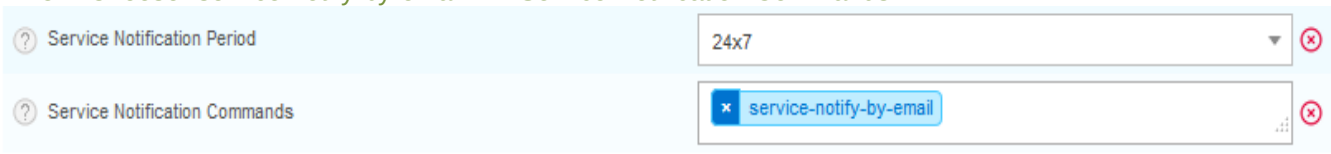

3. Click on your admin user




4. Choose "host-notify-by-email" in "Host Notification Commands"



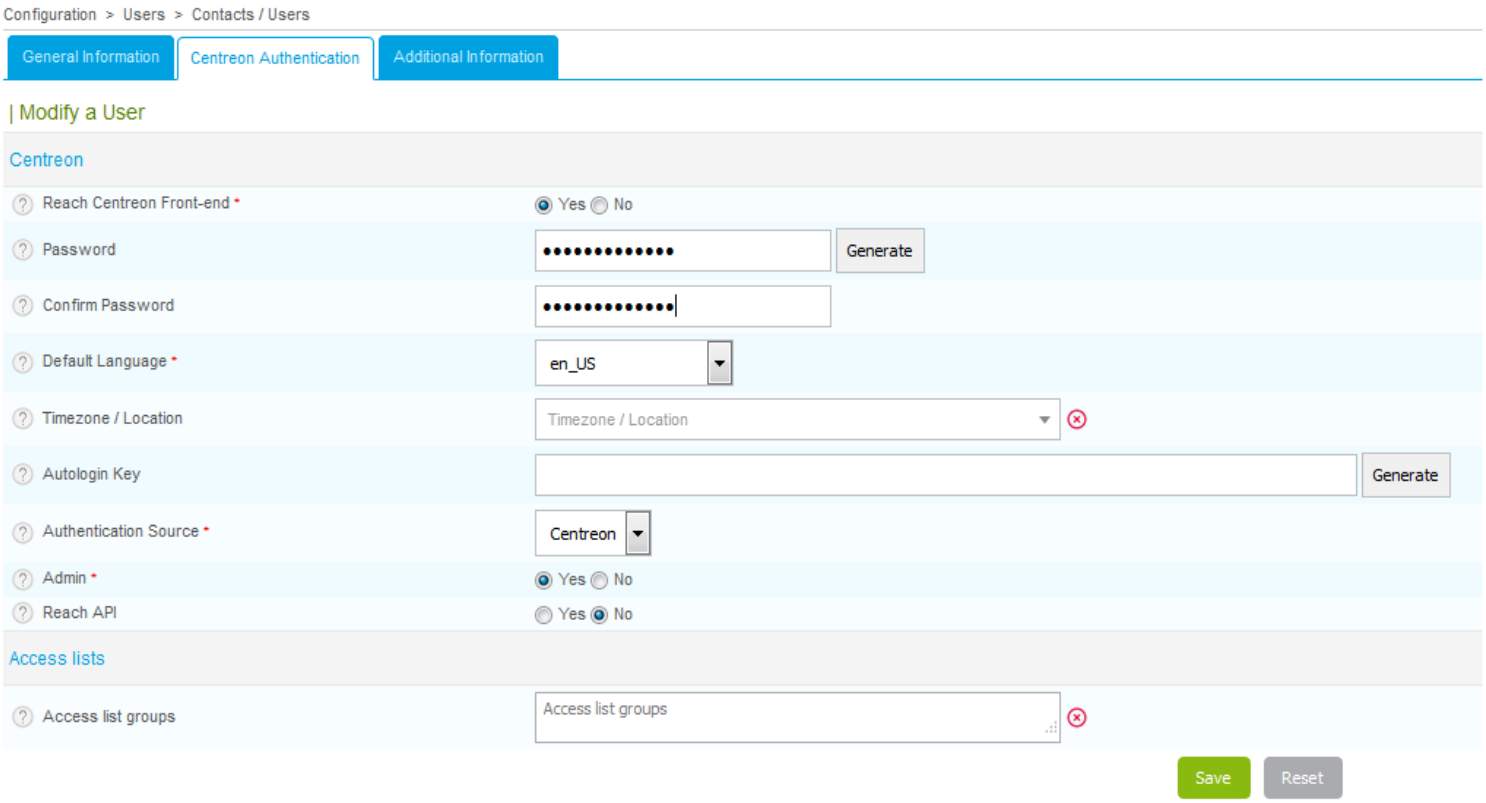

5. Choose "service-notify-by-email" in "Service Notification Commands"




6. Go to "Centreon Authentification"


7. Confirm your password


8. Click on "Save"

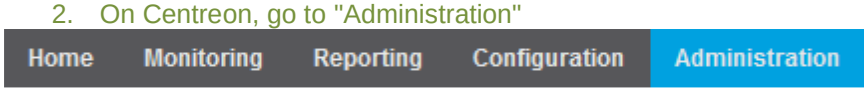



Change the language to French

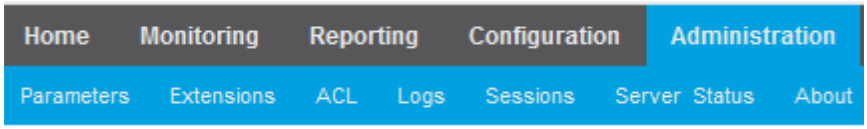
1. Install the language package

```
> yum -y install centreon-lang-fr_FR
```

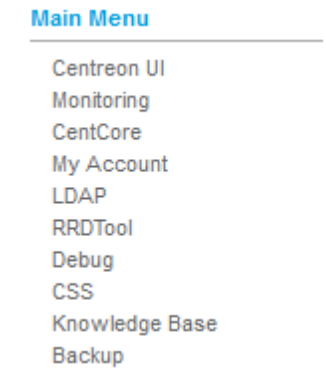
2. On Centreon, go to "Administration"



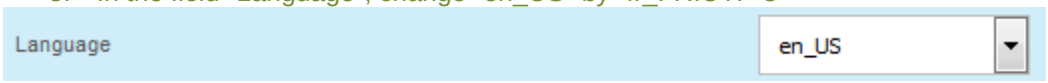

3. Go to "Parameters"



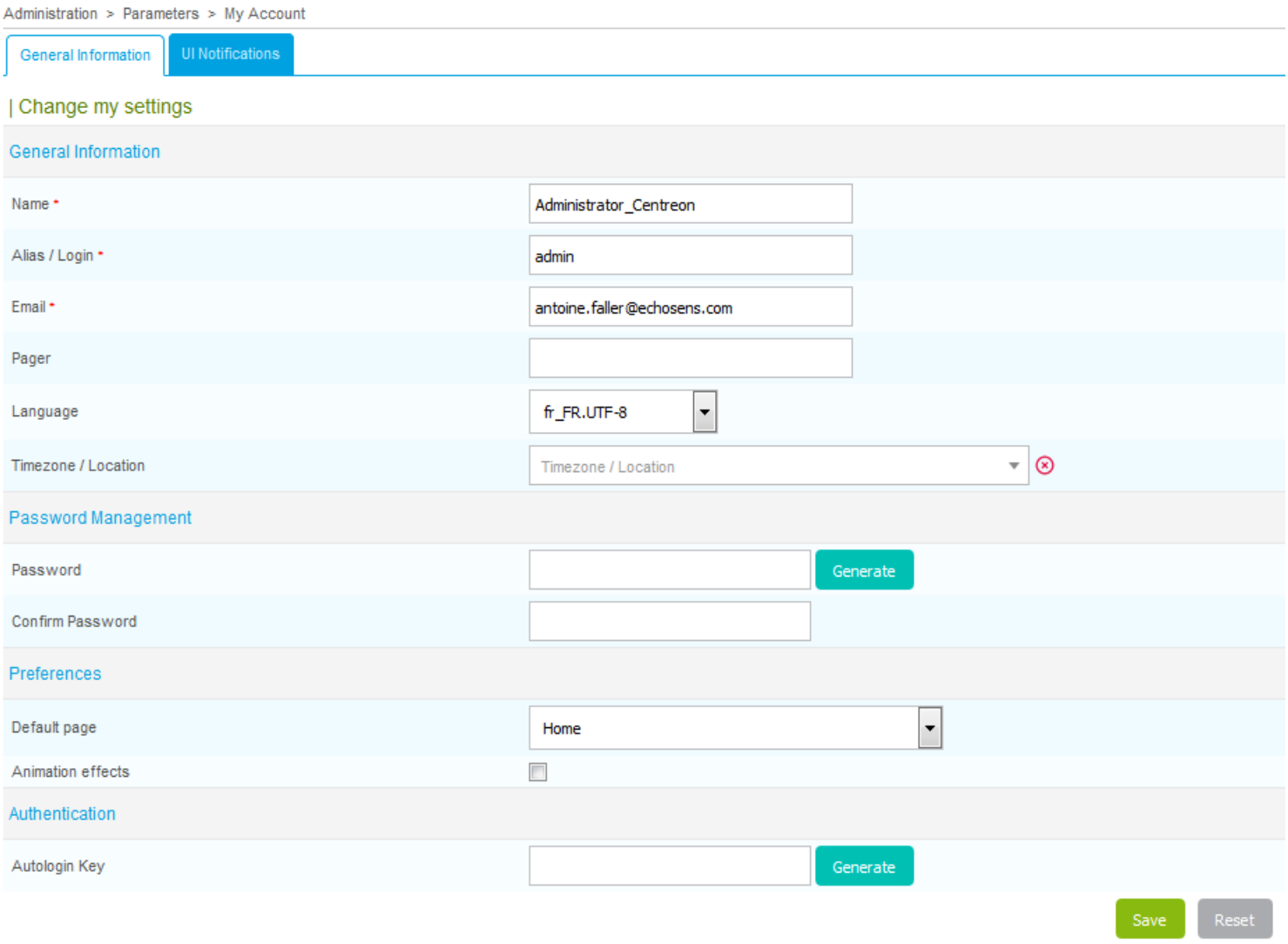

4. In the left menu go to "My Account"




5. In the field "Language", change "en_US" by "fr_FR.UTF-8"

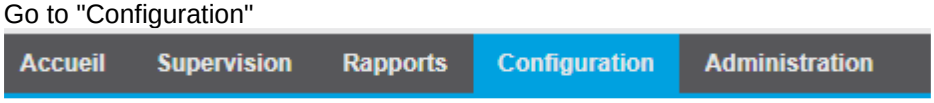



6. Click on "Save"

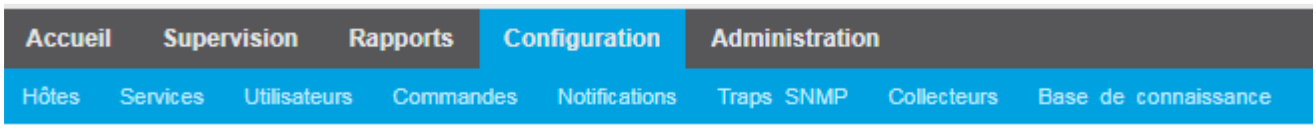



# III. Liaison de NRPE et Centreon

Go to "Configuration"

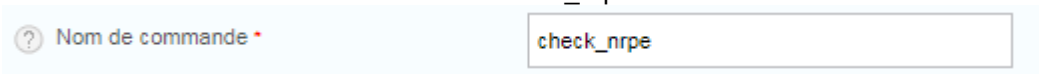



Go to "Commandes"



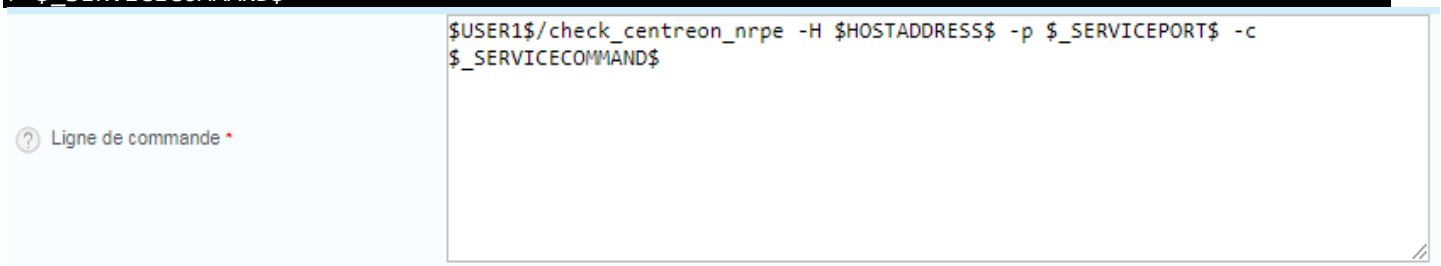

Click on "Ajouter"


Fill the field "Nom de la commande" with "check_nrpe"




Fill the field "Ligne de commande" with :

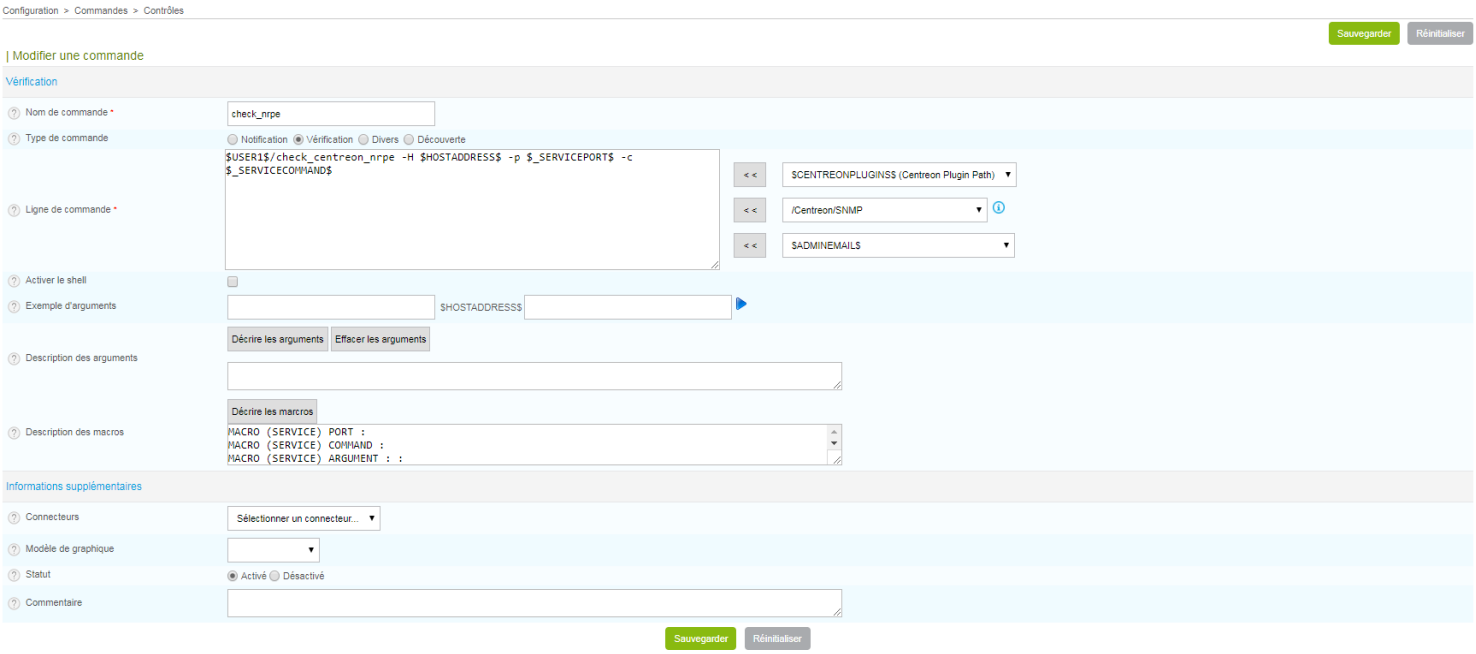
```
> $USER1$/check_centreon_nrpe -H $HOSTADDRESS$ -p $_SERVICEPORT$ -c
> $_SERVICECOMMAND$
```




Fill the field "Description des macros" :

```
MACRO (SERVICE) PORT :
MACRO (SERVICE) COMMAND :
```

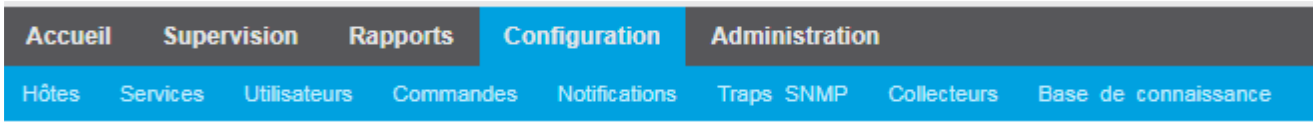

Click on "Sauvegarder"



# IV. Création du modèle NRPE de supervision dans Centreon

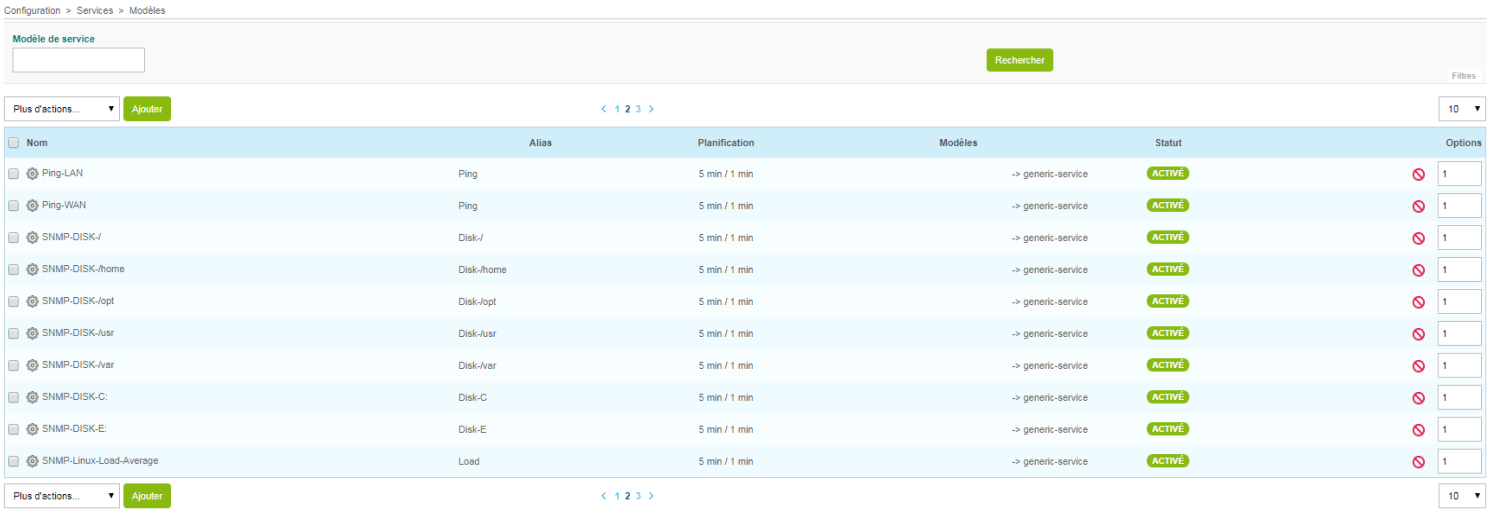
Go to "Configuration"


Go to "Services"



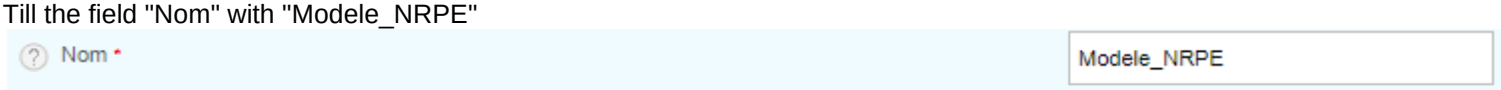

Go to "Modèles"


Click on "Ajouter"

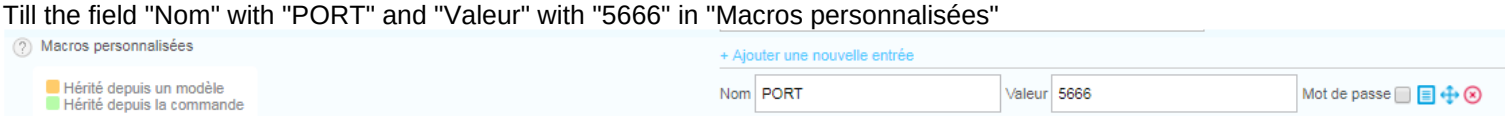



Till the field "Alias" with "NRPE"

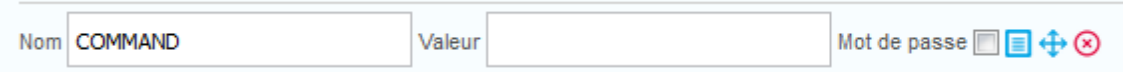

Till the field "Nom" with "Modele_NRPE"



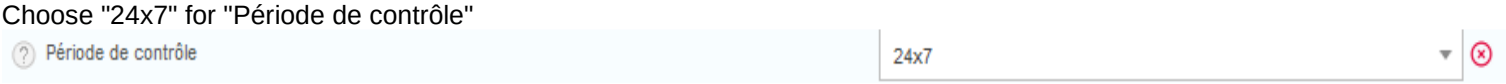

Till the field "Nom" with "PORT" and "Valeur" with "5666" in "Macros personnalisées"



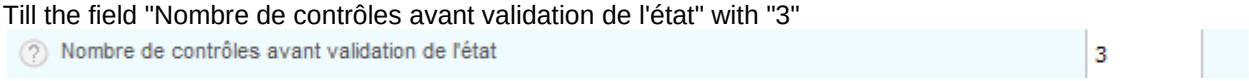

Till the field "Nom" with "COMMAND" and "Valeur" with nothing in "Macros personnalisées"




Choose "24x7" for "Période de contrôle"

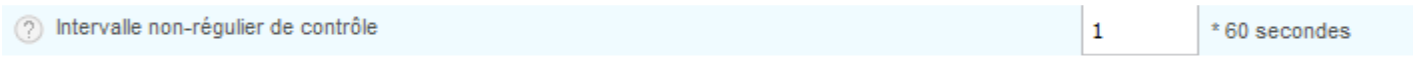



Till the field "Nombre de contrôles avant validation de l'état" with "3"



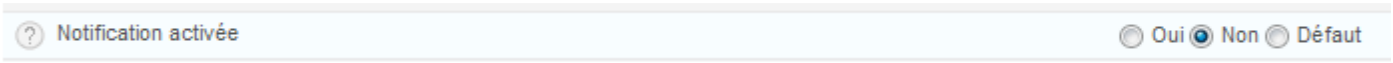

Till the field "Intervalle normal de contrôle" with "3"


Till the field "Intervalle non-régulier de contrôle" with "1"

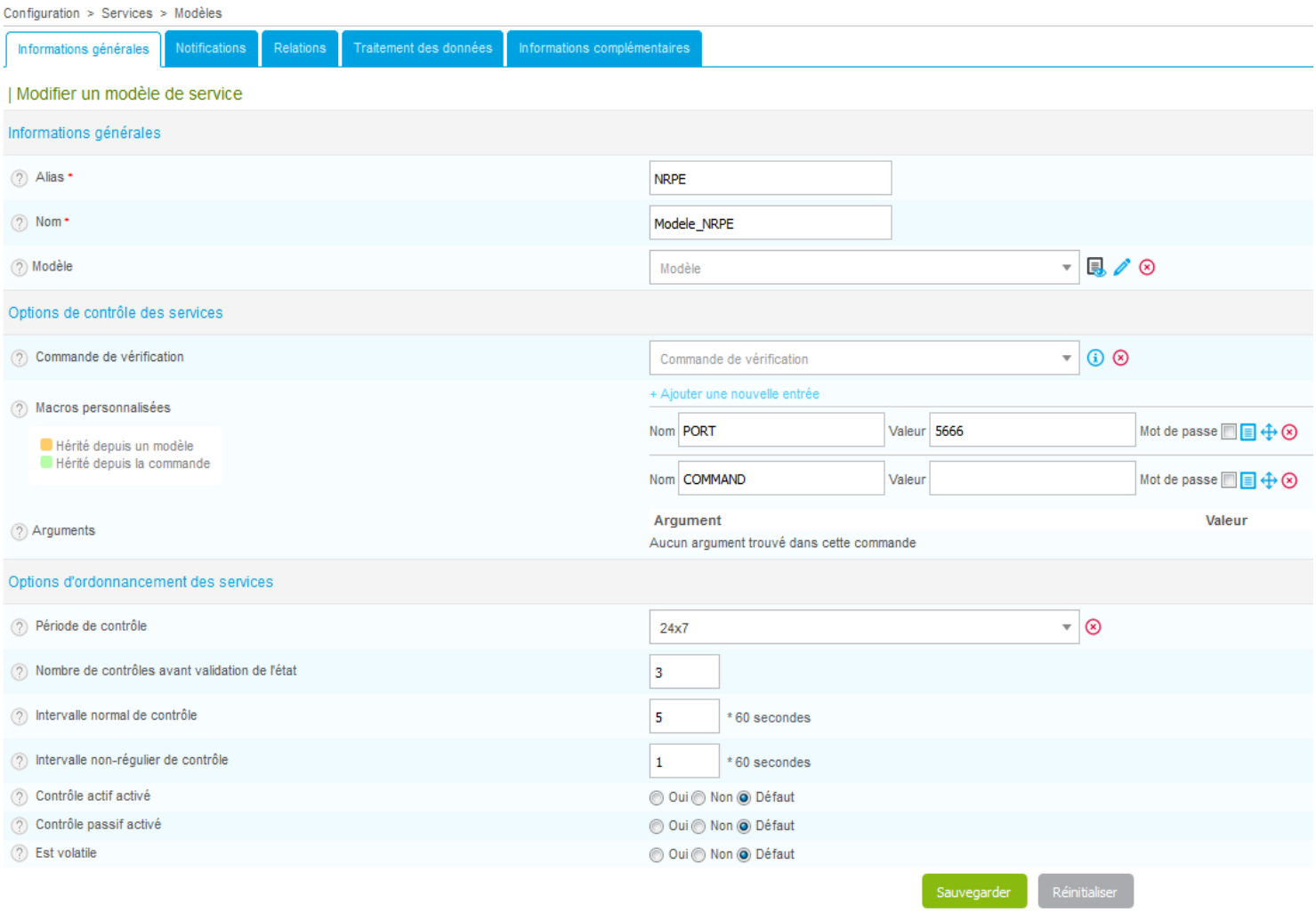



Go to Notifications


Choose "Non" for "Notification activée"

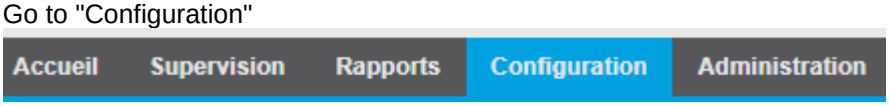



Click on "Sauvegarder"

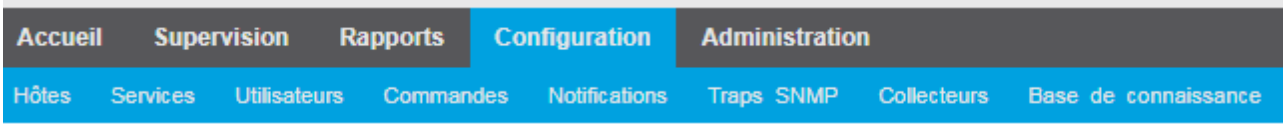



# V. Création d'un hôte dans Centreon

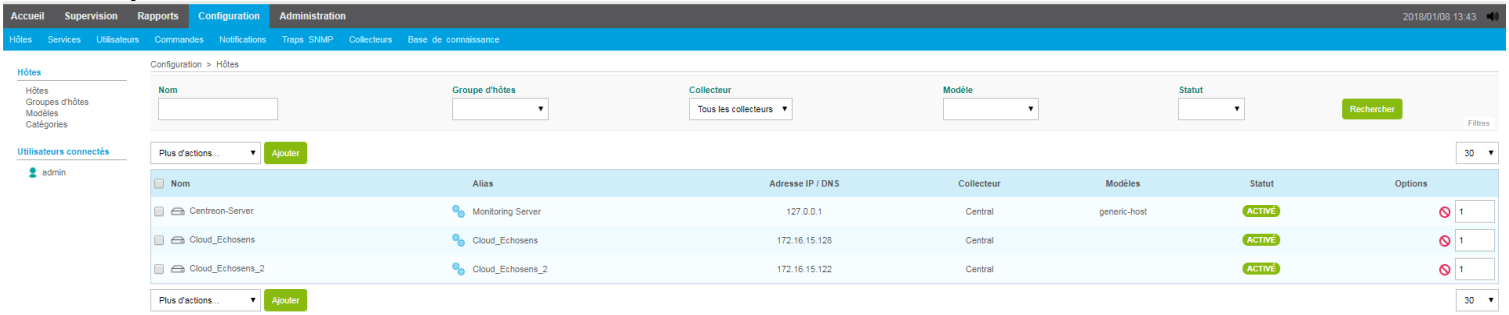
Go to "Configuration"



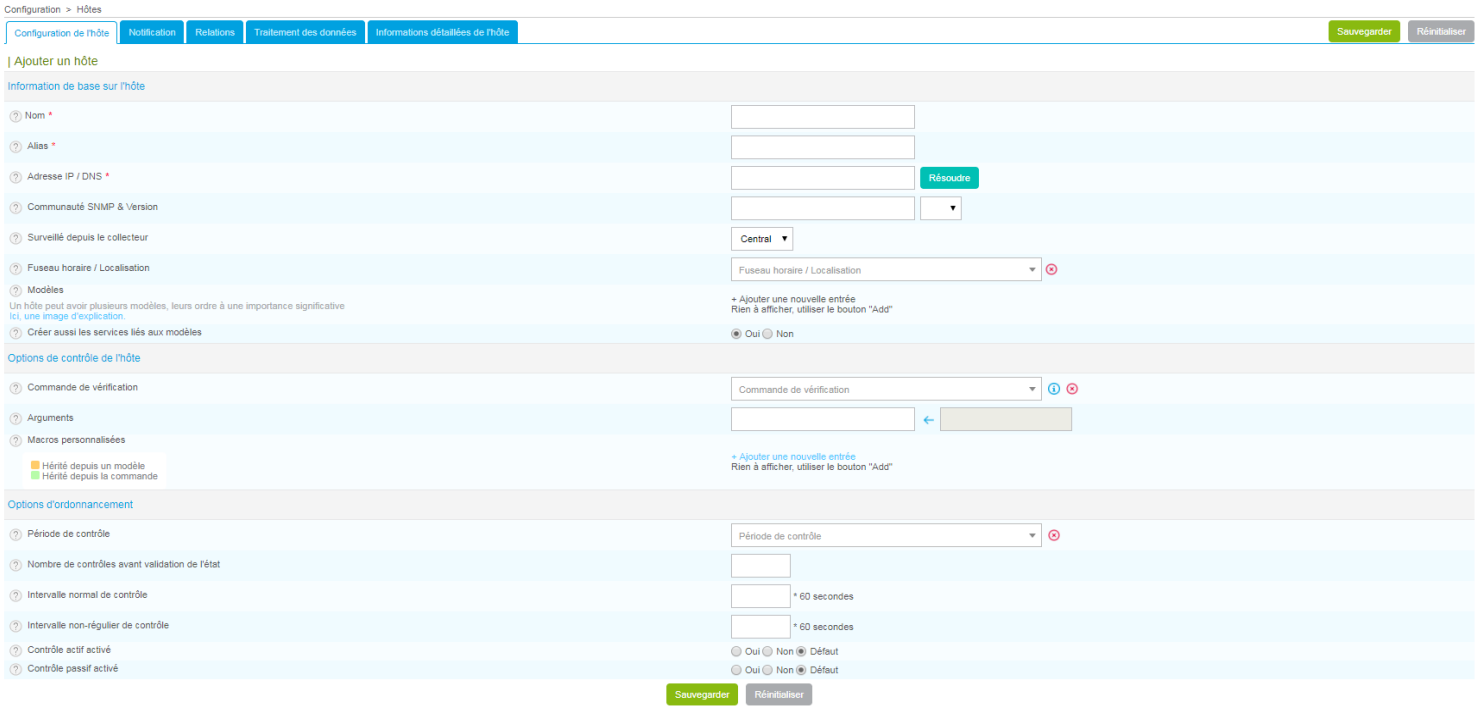

Go to "Hôtes"




Click on "Ajouter"




Fill fields "Nom", "Alias" & "Adresse IP/DNS"



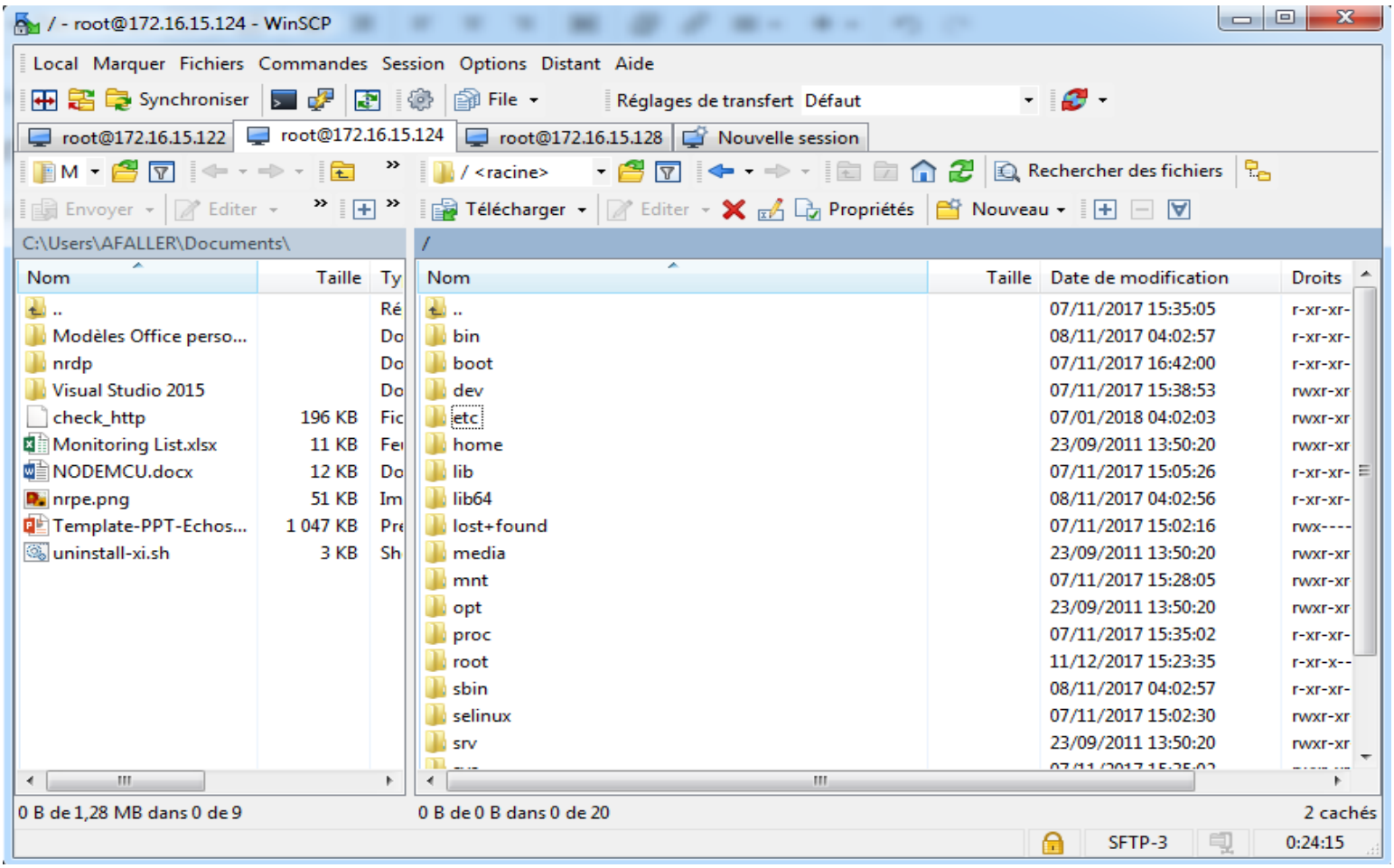

Click on "Sauvegarder"

Your new host is create !

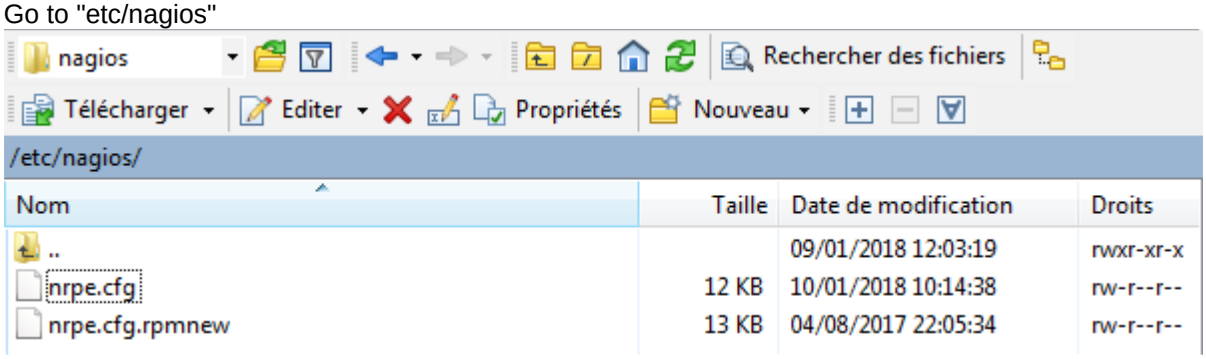
# VI. Configurer les commandes de supervision

Go to etc/nagios/nrpe.cfg

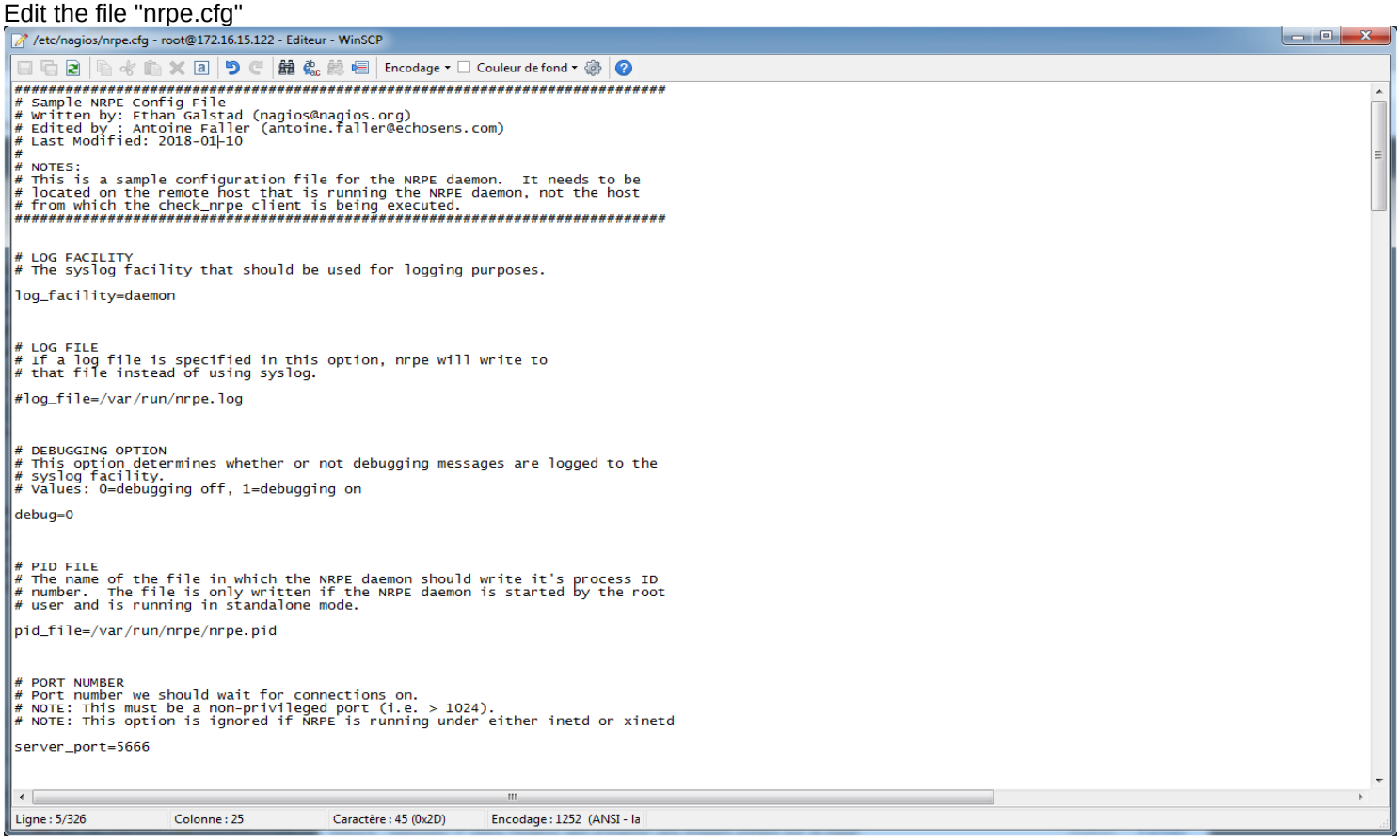
Access the server via WinSCP



Go to "etc/nagios"




Edit the file "nrpe.cfg"




```
##############################################################################
# Sample NRPE Config File
# Written by: Ethan Galstad (nagios@nagios.org)
# Edited by : Antoine Faller (antoine.faller@echosens.com)
# Last Modified: 2018-01-10
#
# NOTES:
# This is a sample configuration file for the NRPE daemon.  It needs to be
# located on the remote host that is running the NRPE daemon, not the host
# from which the check_nrpe client is being executed.
##############################################################################


# LOG FACILITY
# The syslog facility that should be used for logging purposes.

log_facility=daemon



# LOG FILE
# If a log file is specified in this option, nrpe will write to
# that file instead of using syslog.

#log_file=/var/run/nrpe.log



# DEBUGGING OPTION
# This option determines whether or not debugging messages are logged to the
# syslog facility.
# Values: 0=debugging off, 1=debugging on

debug=0



# PID FILE
# The name of the file in which the NRPE daemon should write it's process ID
# number.  The file is only written if the NRPE daemon is started by the root
# user and is running in standalone mode.

pid_file=/var/run/nrpe/nrpe.pid



# PORT NUMBER
# Port number we should wait for connections on.
# NOTE: This must be a non-privileged port (i.e. > 1024).
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd

server_port=5666
```

Go to the line "COMMAND DEFINITIONS" ( ligne 280)

```
# COMMAND DEFINITIONS
# Command definitions that this daemon will run.  Definitions
# are in the following format:
#
# command[<command_name>]=<command_line>
#
# When the daemon receives a request to return the results of <command_name>
# it will execute the command specified by the <command_line> argument.
#
# Unlike Nagios, the command line cannot contain macros - it must be
# typed exactly as it should be executed.
#
# Note: Any plugins that are used in the command lines must reside
# on the machine that this daemon is running on!  The examples below
# assume that you have plugins installed in a /usr/local/nagios/libexec
# directory.  Also note that you will have to modify the definitions below
# to match the argument format the plugins expect.  Remember, these are
# examples only!


# The following examples use hardcoded command arguments...

command[check_users]=/usr/lib64/nagios/plugins/check_users -w 5 -c 10
command[check_load]=/usr/lib64/nagios/plugins/check_load -r -w .15,.10,.05 -c .30,.25,.20
command[check_hda1]=/usr/lib64/nagios/plugins/check_disk -w 20% -c 10% -p /dev/hda1
command[check_zombie_procs]=/usr/lib64/nagios/plugins/check_procs -w 5 -c 10 -s Z
command[check_total_procs]=/usr/lib64/nagios/plugins/check_procs -w 150 -c 200
command[check_wildfly_procs]=/usr/lib64/nagios/plugins/check_procs -w 3: -c 1: -u wildfly
command[check_java_procs]=/usr/lib64/nagios/plugins/check_procs -c 1: -C java
command[check_mysql_procs]=/usr/lib64/nagios/plugins/check_procs -c 1: -u mysql
command[check_apache_procs]=/usr/lib64/nagios/plugins/check_procs -w 5: -c 1: -u apache
command[check_wildfly_http]=/usr/lib64/nagios/plugins/check_http -I localhost -p 8080
command[check_sso_client_login_http]=/usr/lib64/nagios/plugins/check_http -H 172.16.15.122 -p 9001 -u /client/login -T "application/json" -j POST -P '{"clientId
command[check_sso_client_login_http_url]=/usr/lib64/nagios/plugins/check_http -I 172.16.15.125 -p 443 -u https://discovery.dev-echosens.fr/login -S -j GET



# The following examples allow user-supplied arguments and can
# only be used if the NRPE daemon was compiled with support for
# command arguments *AND* the dont_blame_nrpe directive in this
# config file is set to '1'.  This poses a potential security risk, so
# make sure you read the SECURITY file before doing this.

#command[check_users]=/usr/lib64/nagios/plugins/check_users -w $ARG1$ -c $ARG2$
#command[check_load]=/usr/lib64/nagios/plugins/check_load -w $ARG1$ -c $ARG2$
#command[check_disk]=/usr/lib64/nagios/plugins/check_disk -w $ARG1$ -c $ARG2$ -p $ARG3$
#command[check_procs]=/usr/lib64/nagios/plugins/check_procs -w $ARG1$ -c $ARG2$ -s $ARG3$
```

Type below the new commands that you would like to add
You can learn more on : https://www.monitoring-plugins.org/doc/man/

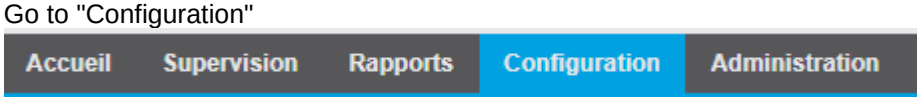
Once the modifications are finished , save the file
Open a terminal and execute the below command :

```
> service nrpe restart
```

The newly added commands are now available via your Centreon server

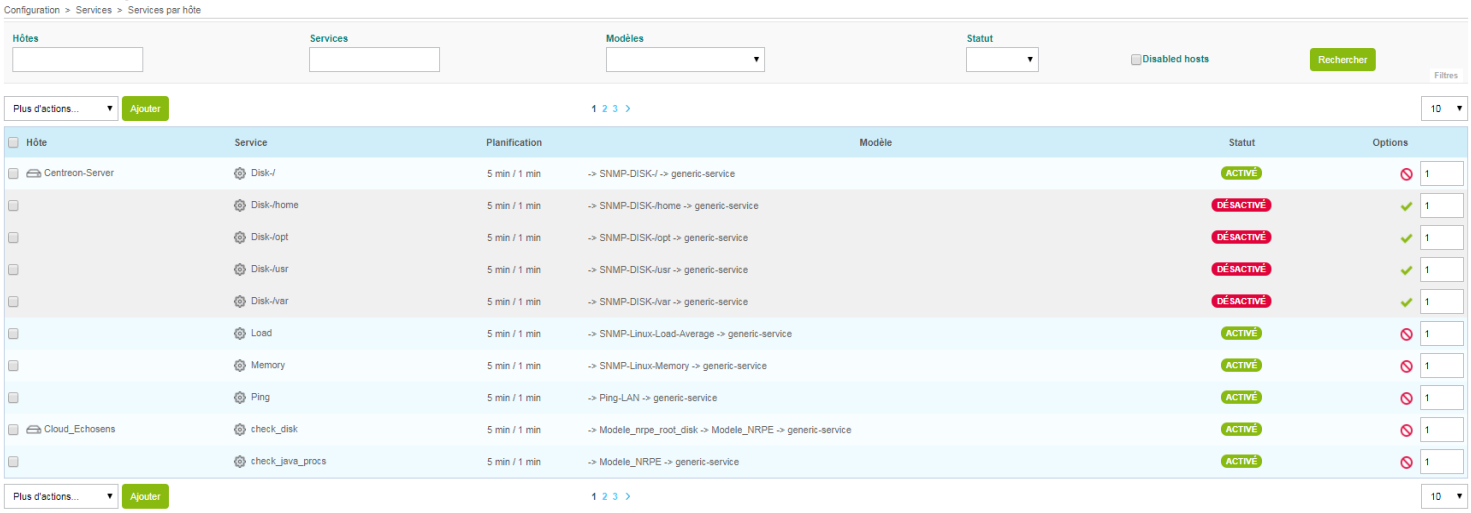
# VII. Création d'un service dans Centreon

Go to "Configuration"

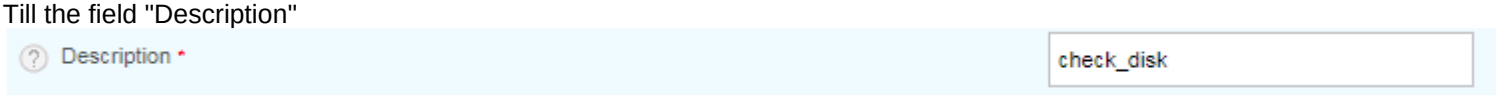



Go to "Services"

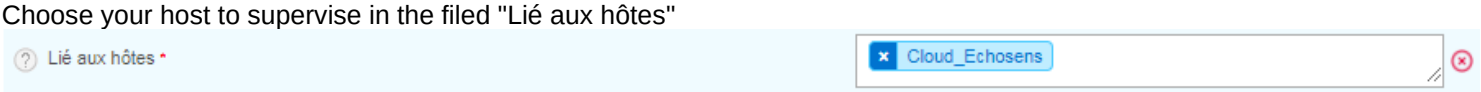

Click on "Ajouter"




Till the field "Description"

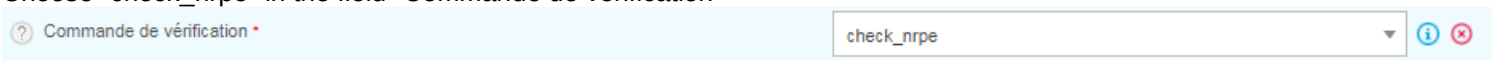



Choose your host to supervise in the filed "Lié aux hôtes"

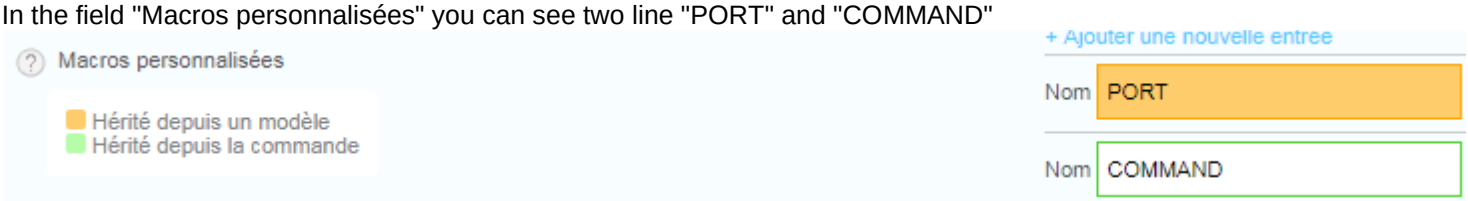



Choose "Modele_NRPE" in the field "Modèle"

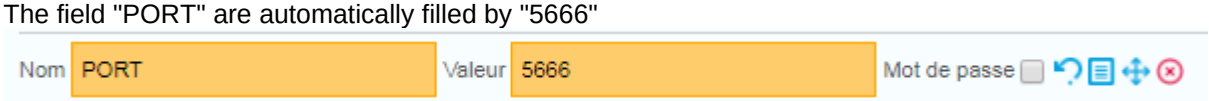

Choose "check_nrpe" in the field "Commande de vérification"



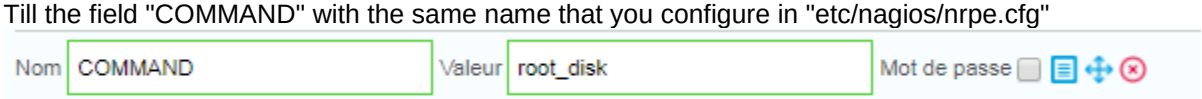

In the field "Macros personnalisées" you can see two line "PORT" and "COMMAND"



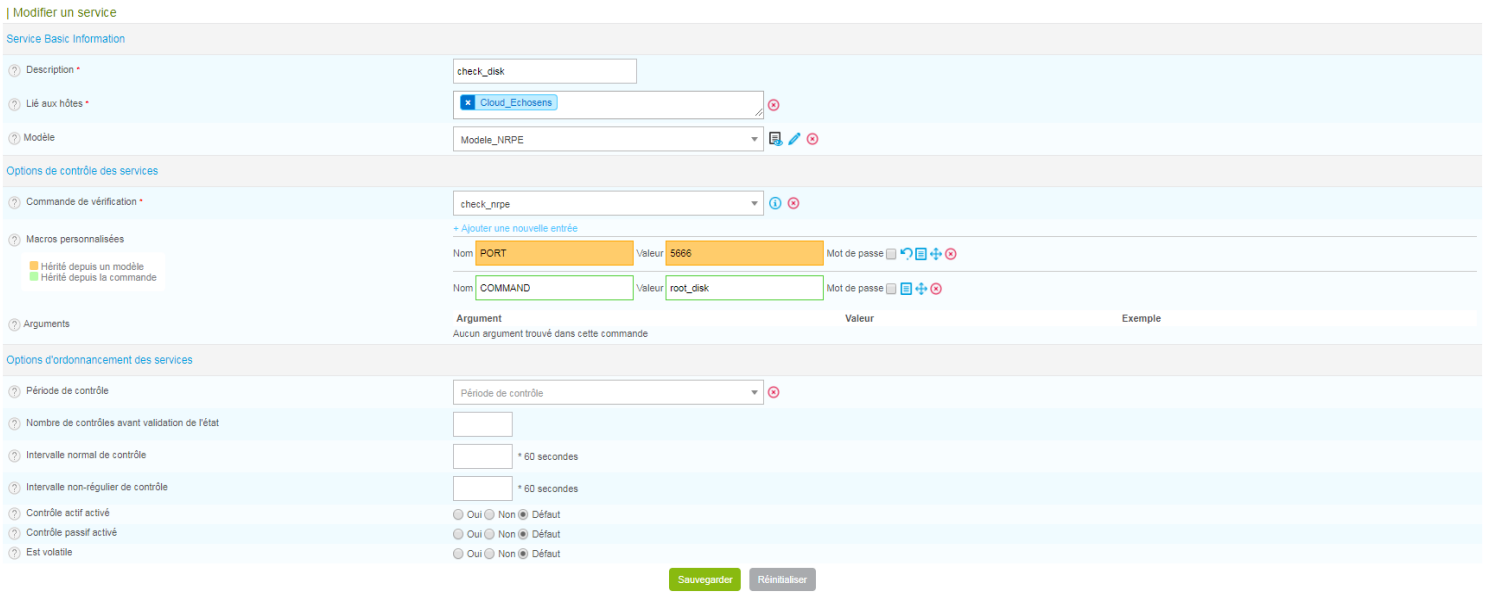

The field "PORT" are automatically filled by "5666"




Till the field "COMMAND" with the same name that you configure in "etc/nagios/nrpe.cfg"

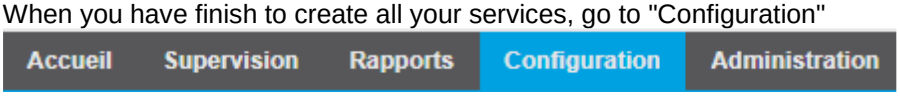



Click on "Sauvegarder"

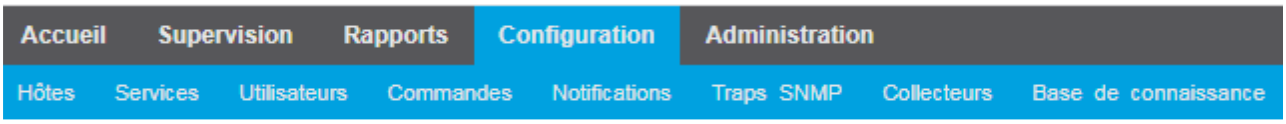



# VIII. Affichage de la supervision

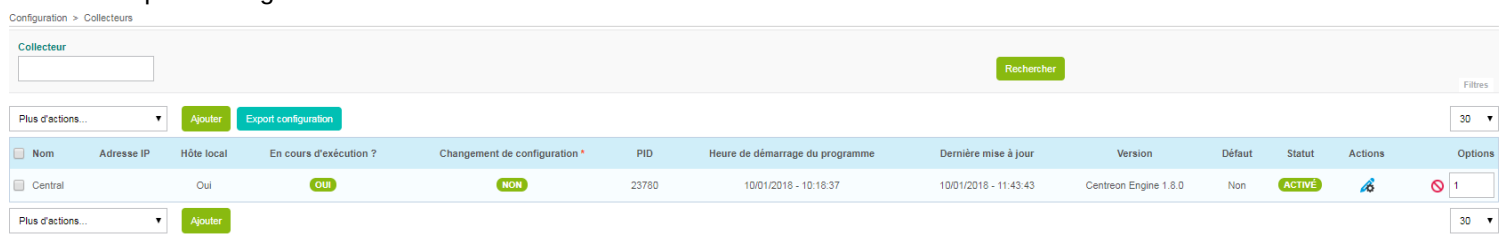
When you have finish to create all your services, go to "Configuration"



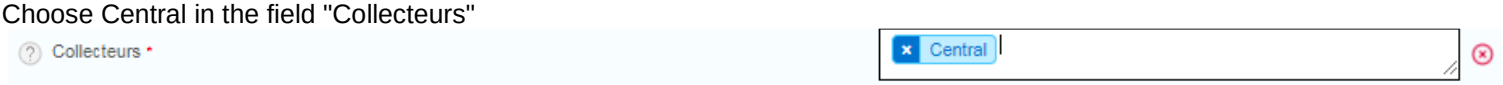

Go to "Collecteurs"




Click on "Export Configuration"

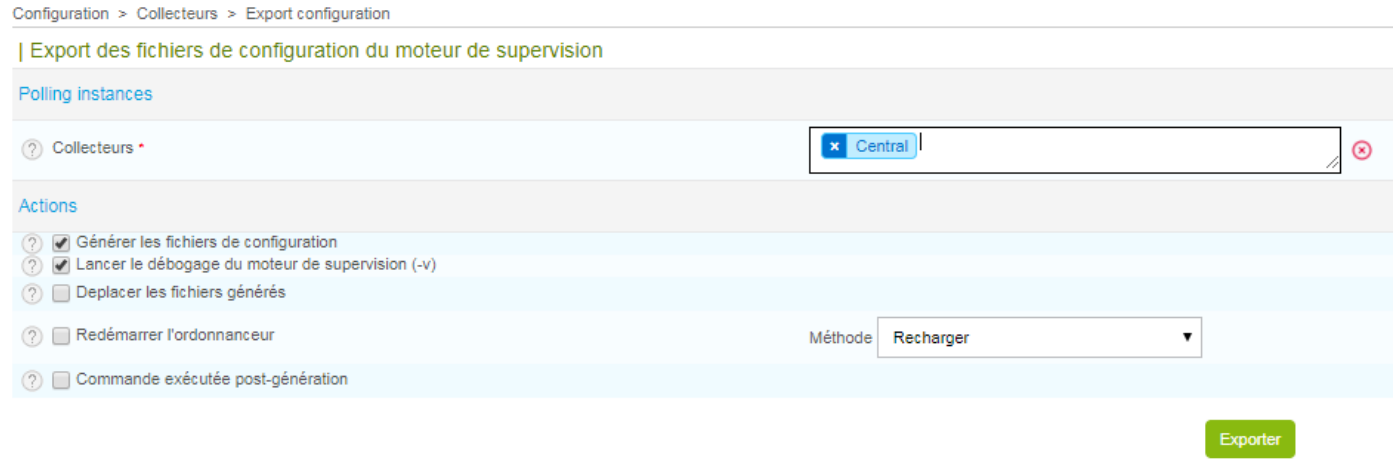



Choose Central in the field "Collecteurs"




Check the two first checkbox : "Générer les fichiers de configuration" & "Lancer le débogage du moteur de supervision (-v)"

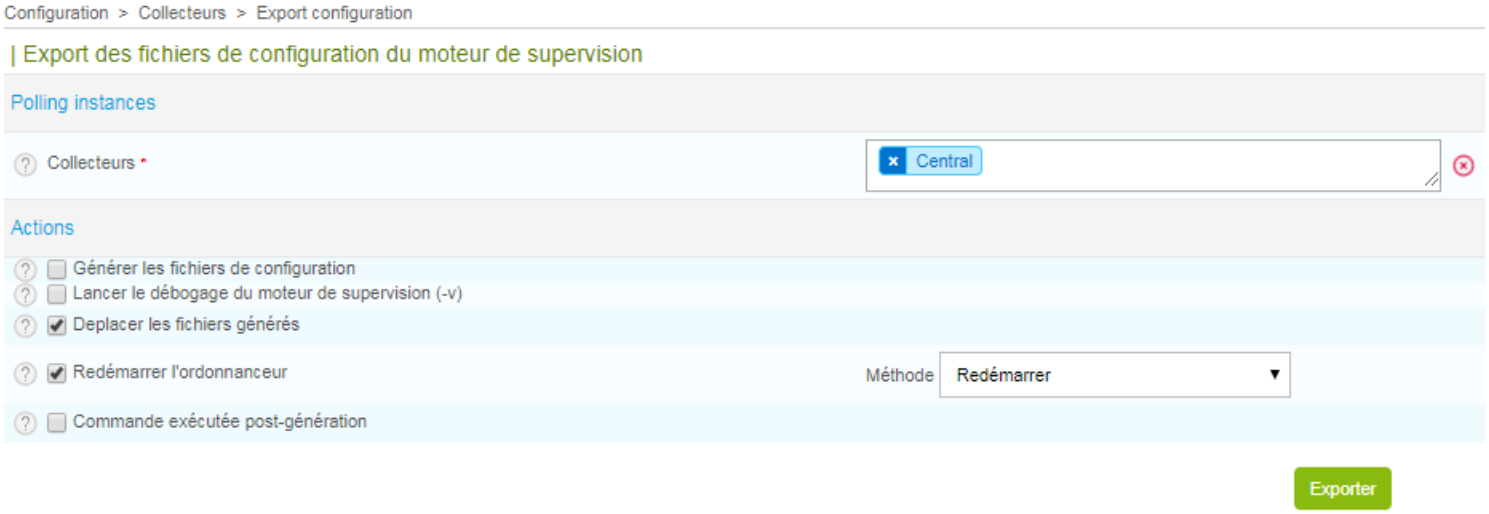

Click on "Export"



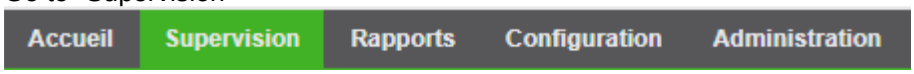
Uncheck the two first checkbox and check the two next : "Déplacer les fichiers générés" & "Redémarrer l'ordonnanceur"


Change "Recharger" in "Méthode" by "Redémarrer"

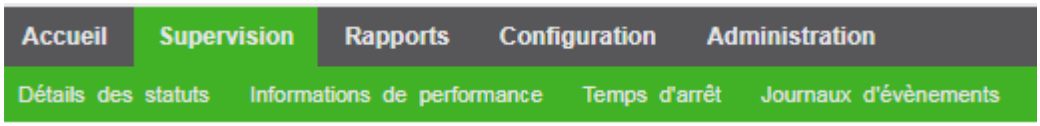

Click on "Export"



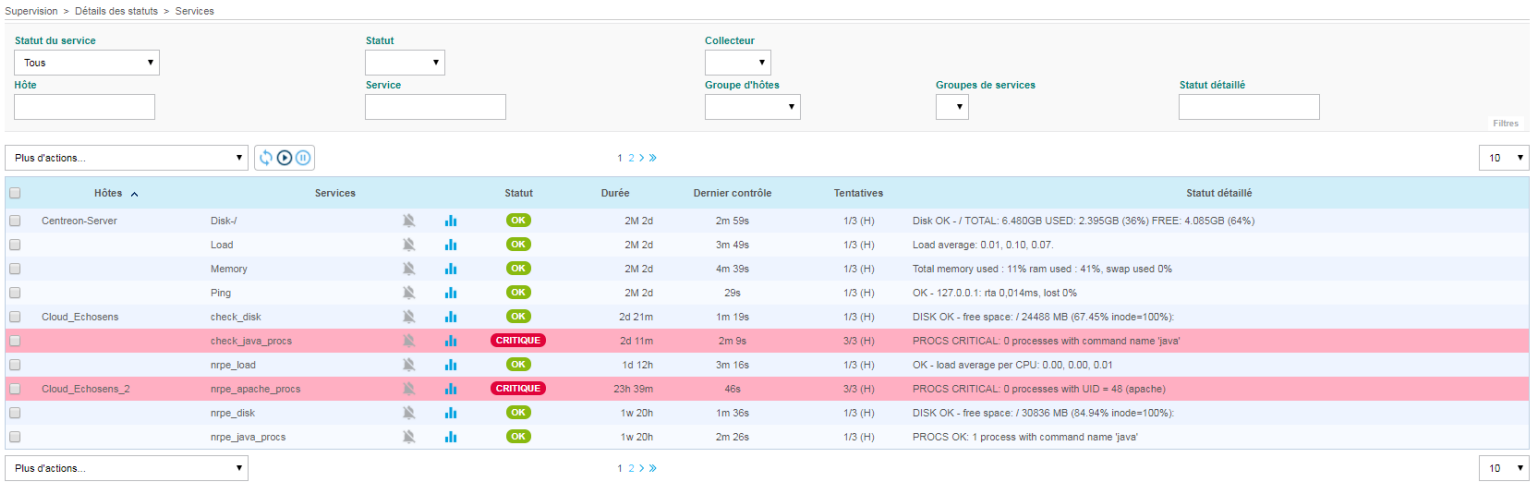

Go to "Supervision"




Go to "Détails des statuts"




Choose "Tous" in "Statut du service"




Congratulations ! Your supervision is ready !