Table des matières

[1. Création des commandes 1](#_Toc92439798)

[2. Création des templates de services 12](#_Toc92439799)

[3. Création des templates d’hôtes 20](#_Toc92439800)

[4. Ajout des hôtes dans le serveur 21](#_Toc92439801)

[5. Application des modifications sur le serveur 23](#_Toc92439802)

[6. Monitoring des hôtes 23](#_Toc92439803)

# 1. Création des commandes

**Machine Windows 10 (NRPE) :**

Pour les différentes commandes en NRPE nous pouvons trouver les infos sur le site suivant :

https://docs.nsclient.org/reference/

Il faut aller dans l’onglet références -> Check commands et dedans nous avons les différents modules installés avec NSClient++

Dans ces modules nous pouvons trouver les commandes à passer ci-dessous (surligné en vert) pour vérifier ce que l’on a besoin sur la machine distante

En cliquant sur la commande on peut accéder à des exemples d’utilisation avec des syntaxes

**Vérification du stockage sur le disque C en commande NRPE :**

On commence par trouver quelle sonde nous avons besoin et nous la testons en ligne de commande :

Seuil de Warning à 80% d’occupation

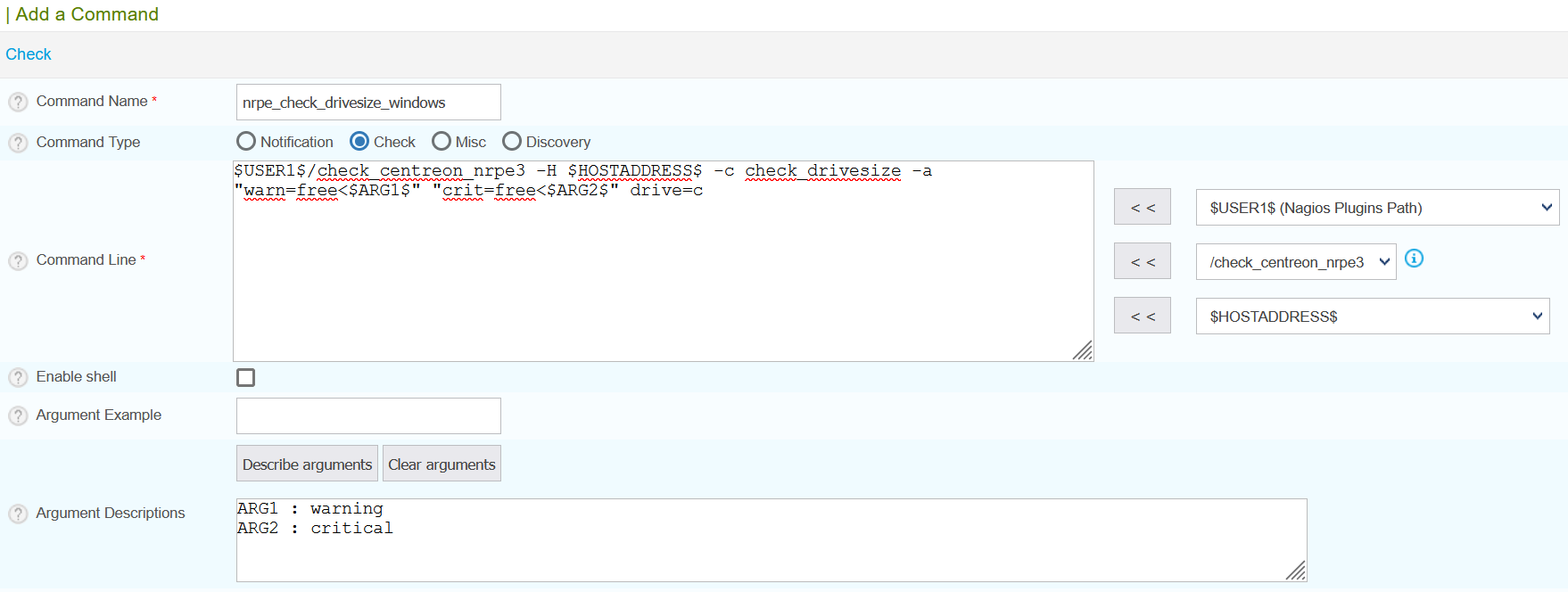
Seuil Critique à 90% d’occupation

[root@localhost libexec]# /usr/lib64/nagios/plugins/check\_centreon\_nrpe3 -H 192.168.6.1 -c check\_drivesize -a "warn=free<20%" "crit=free<10%" drive=c

CHECK\_NRPE: Invalid packet version received from server.

OK All 1 drive(s) are ok|'c: free'=7.60047GB;3.88743;1.94371;0;19.43718 'c: free %'=39%;20;10;0;100

Maintenant que la commande fonctionne bien nous devons la créer dans notre serveur de supervision :



La commande est maintenant créée

**Vérification de la quantité de mémoire en commande NRPE :**

Seuil de Warning à 80% d’occupation

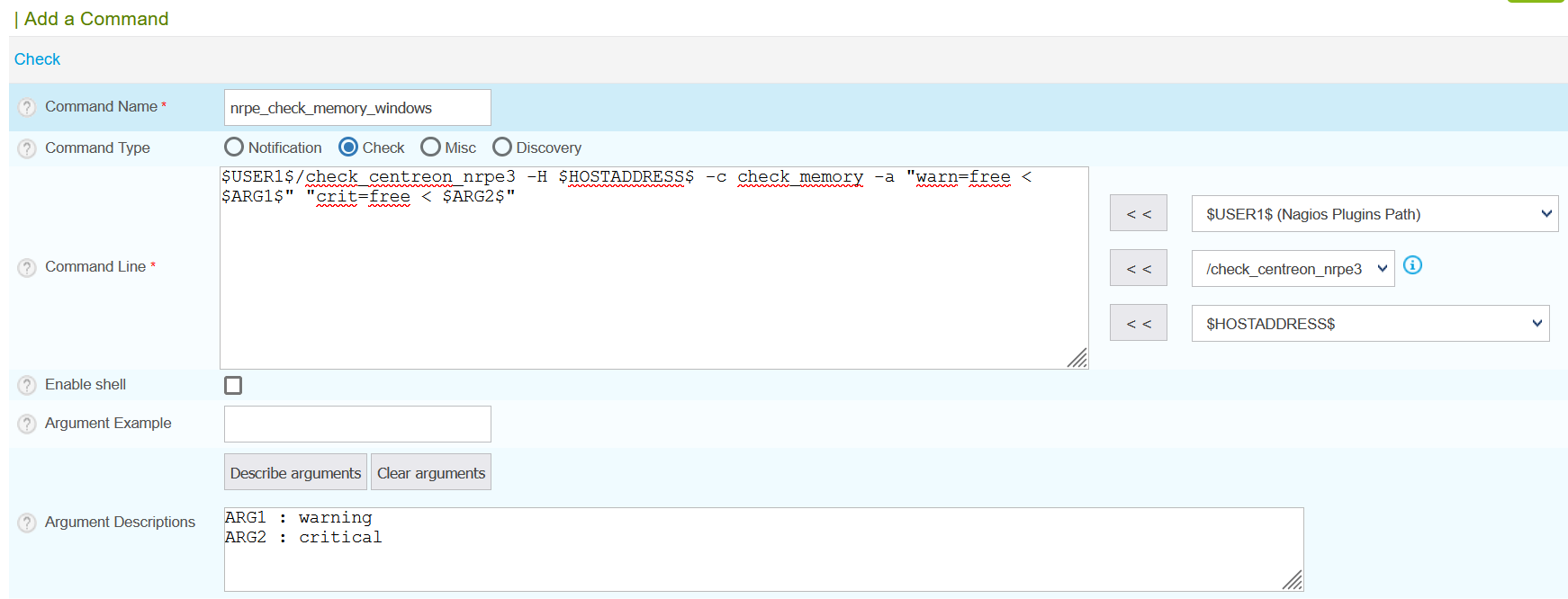
Seuil Critique à 90% d’occupation

[root@localhost libexec]# /usr/lib64/nagios/plugins/check\_centreon\_nrpe3 -H 192.168.6.1 -c check\_memory -a "warn=free < 20%" "crit=free < 10%"

CHECK\_NRPE: Invalid packet version received from server.

OK: committed = 1.343GB, physical = 1.184GB|'committed'=1.78181GB;0.62491;0.31245;0;3.12456 'committed %'=57%;20;10;0;100 'physical'=835.28906MB;409.51015;204.75507;0;2047.55078 'physical %'=41%;20;10;0;100

Maintenant que la commande fonctionne bien nous devons la créer dans notre serveur de supervision :



**Vérification de la charge CPU en commande NRPE :**

Seuil de Warning à 80% d’occupation

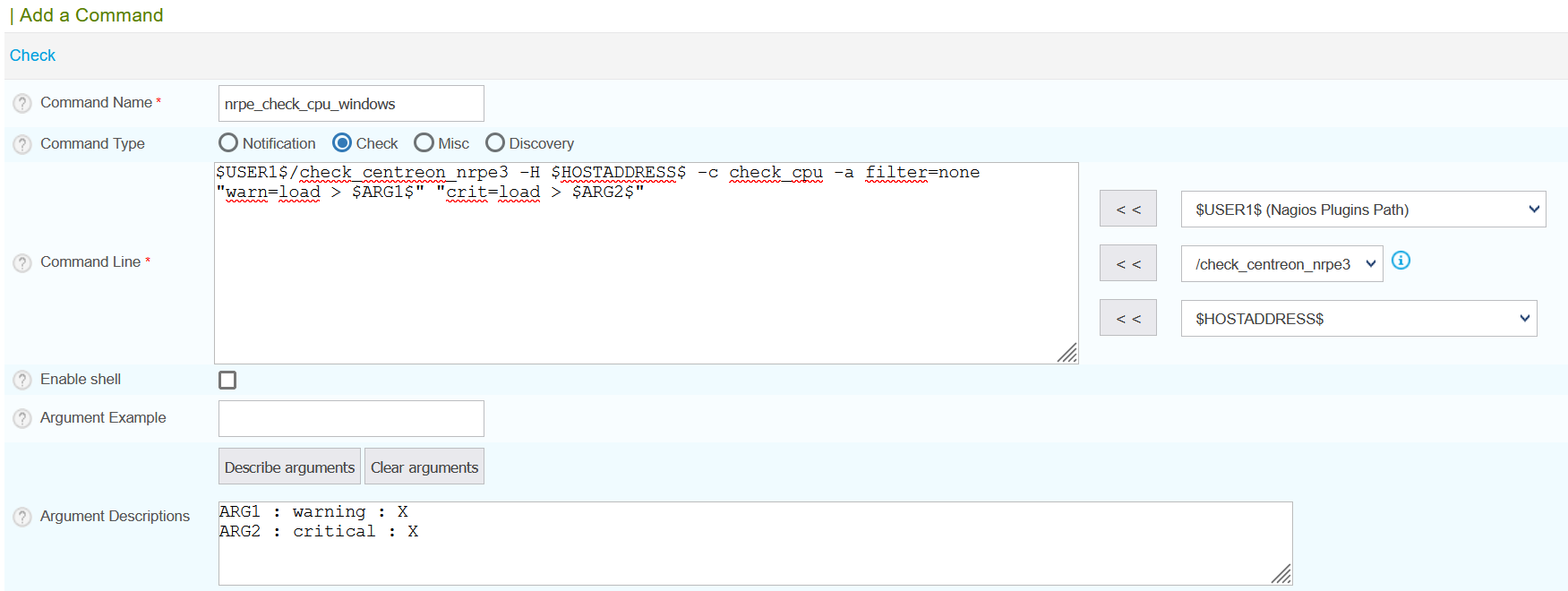
Seuil Critique à 90% d’occupation

[root@localhost libexec]# /usr/lib64/nagios/plugins/check\_centreon\_nrpe3 -H 192.168.6.1 -c check\_cpu -a filter=none "warn=load > 80" "crit=load > 90"

CHECK\_NRPE: Invalid packet version received from server.

OK: CPU load is ok.|'core 0 5m'=0%;80;90 'total 5m'=0%;80;90 'core 0 1m'=0%;80;90 'total 1m'=0%;80;90 'core 0 5s'=1%;80;90 'total 5s'=1%;80;90

Maintenant que la commande fonctionne bien nous devons la créer dans notre serveur de supervision :



**Vérification des mises à jour en commande NRPE :**

Il n’existe pas de plugin permettant de vérifier si des mises à jour sont disponibles. Il faut donc télécharger ou créer un script qui permettra de répondre aux attendus

Pour se faire nous nous sommes aidés du site suivant :

https://exchange.nagios.org/directory/Plugins/Operating-Systems/Windows-NRPE/Check-Windows-Updates-using-Powershell/details

Sur ce site nous devons télécharger le script donné et le déplacer dans le répertoire "scripts" de notre NSClient :



Une fois le script mis dans le bon répertoire il faut lancer un Powershell en mode administrateur et lancer la commande suivante pour passer la sécurité :

set-executionpolicy bypass

Il faut vérifier dans le nsclient.ini que le module "CheckExternalScripts" est bien activé :

CheckExternalScripts = enabled

Ensuite dans le même fichier nsclient.ini y ajouter les lignes suivantes :

[/settings/external scripts/scripts]

check\_updates=cmd /c echo scripts\check\_windows\_updates.ps1; exit $LastExitCode | powershell.exe -command –

La 1ère ligne fait le lien vers l’emplacement de notre script et la 2ème ligne permet d’associer une commande au lancement du script

Maintenant que tout est bien configuré il faut redémarrer le service nsclient dans un cmd en mode administrateur :

cd "\Program Files\NSClient++"

net stop nscp && net start nscp

Nous pouvons maintenant passer la commande de check\_update en mode NRPE :

[root@localhost libexec]# /usr/lib64/nagios/plugins/check\_centreon\_nrpe3 -H 192.168.6.1 -c check\_updates

CHECK\_NRPE: Invalid packet version received from server.

Updates: 3 critical, 1 optional

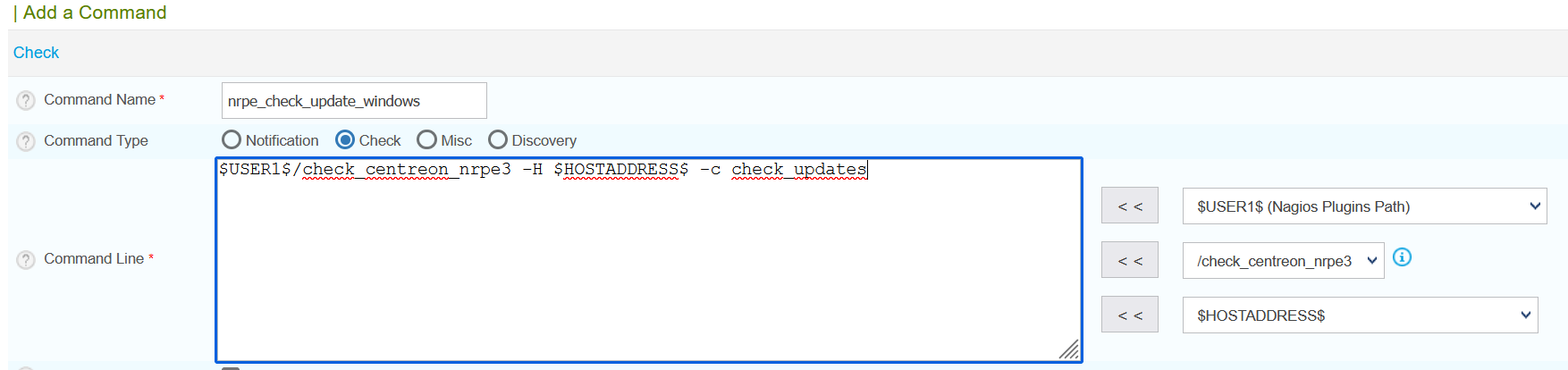
Outil de suppression de logiciels malveillants Windows x64▒- v5.96 (KB890830)

2021-11 Mise ▒ jour pour Windows 10 Version 21H2 sur syst▒mes x64 (KB4023057)

2021-12 Mise ▒ jour cumulative pour Windows 10 Version 21H2 pour les syst▒mes x64 (KB5008212)

|'critical'=3;0;0

Maintenant que la commande fonctionne bien nous devons la créer dans notre serveur de supervision :



**Machine Debian (SNMP & SSH) :**

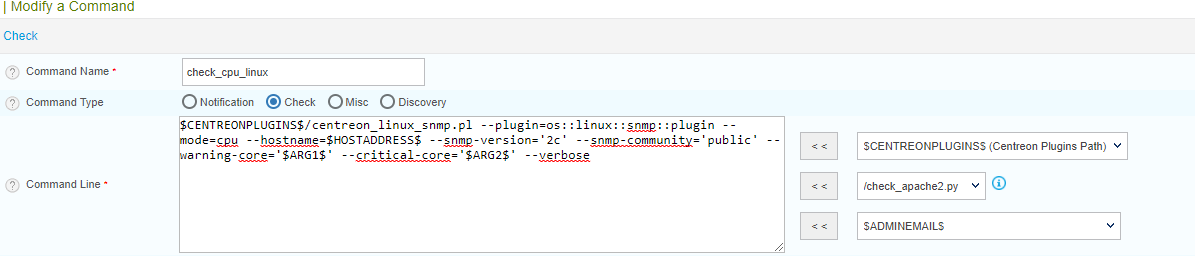
**Partie SNMP :**

**Vérification de la charge CPU via SNMP avec taux de warning à 80% et taux de criticité à 90% :**

Ligne de commande :

/usr/lib/centreon/plugins/centreon\_windows\_snmp.pl --plugin=os::windows::snmp::plugin --mode=cpu --hostname=192.168.6.1 --snmp-version='2c' --snmp-community='public' --warning-core='80' --critical-core='90' --verbose

Création de la commande sur centreon :

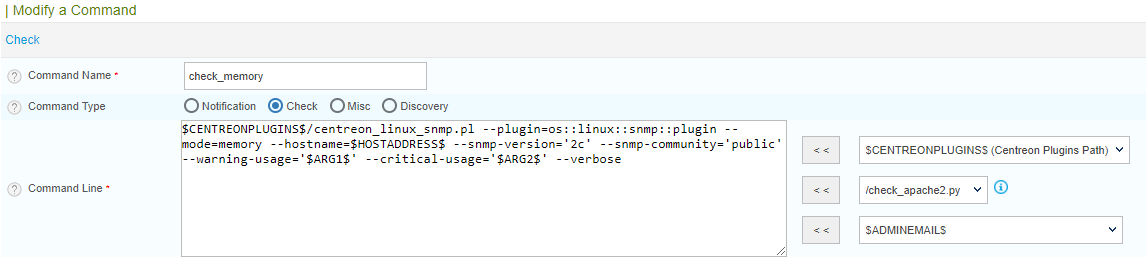


**Vérification de la RAM via SNMP avec taux de warning à 80% et taux de criticité à 90% :**

Ligne de commande :

/usr/lib/centreon/plugins//centreon\_linux\_snmp.pl --plugin=os::linux::snmp::plugin --mode=memory --hostname=192.168.7.1 --snmp-version='2c' --snmp-community='public' --warning-usage='80' --critical-usage='90' –verbose

Création de la commande sur centreon :



**Vérification du stockage sur le disque C via SNMP avec taux de warning à 80% et taux de criticité à 90% :**

Ligne de commande :

/usr/lib/centreon/plugins//centreon\_linux\_snmp.pl --plugin=os::linux::snmp::plugin --mode=storage --hostname=192.168.7.1 --snmp-version='2c' --snmp-community='public' --warning-usage='80' --critical-usage='90' –verbose

/usr/lib64/nagios/plugins/check\_disk.sh -H 192.168.7.1 -C public -i 1 -w 0:80 -c 0:90 -l "/"

Création des commandes sur centreon :

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

**Vérification du bon fonctionnement d’Apache via un test de connexion au port avec le protocole TCP :**

Ligne de commande :

/usr/lib64/nagios/plugins/check\_tcp -H 192.168.7.1 -p 80

Création de la commande sur centreon :

Une image contenant texte

Description générée automatiquement

Passons désormais aux commandes permettant d’établir la supervision de notre machine Windows Server.

**Partie SSH :**

Le principe du check by ssh est que le serveur va se connecter à une machine distante et va exécuter un plugin en local afin de superviser la machine

Pour effectuer un check by ssh nous devons d’abord installer les plugins nagios en local sur notre machine debian :

apt install wget gcc

cd /tmp

wget https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz

tar -xvf nagios-plugins-2.3.3.tar.gz

cd nagios-plugins-2.3.3

./configure

make

make install

Une fois les plugins installés sur notre machine debian nous devons générer une paire de clés publique/privée ssh sur notre serveur de supervision et nous allons copier la clé publique sur la machine debian

Cette paire de clé va nous permettre de ne pas donner de mot de passe lors de la connexion ssh :

Connexion à l’utilisateur centreon-engine :

[root@localhost .ssh]# su - centreon-engine

Dernière connexion : vendredi 7 janvier 2022 à 02:40:44 EST sur pts/0

Génération de la paire de clés :

**/!\ Nous n’entrons pas de passphrase pour pouvoir passer la commande directement sans avoir besoin de mettre de mot de passe**

[centreon-engine@localhost ~]$ ssh-keygen -t rsa -b 4096

Generating public/private rsa key pair.

Enter file in which to save the key (/var/lib/centreon-engine/.ssh/id\_rsa):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /var/lib/centreon-engine/.ssh/id\_rsa.

Your public key has been saved in /var/lib/centreon-engine/.ssh/id\_rsa.pub.

The key fingerprint is:

SHA256:Su8DQ23nnoIEqn5On997qlql2jf3mRVz1CVe37e3zxQ centreon-engine@localhost.localdomain

The key's randomart image is:

+---[RSA 4096]----+

| . o|

| . o=|

| . . \*|

| . . o . .o|

| . o..S+ Eo|

| . .+oo . B|

| . . ..\*.. . o.|

|. ... \*.+ooo +o.|

|..o. =o++\*\* .+ o|

+----[SHA256]-----+

Copie de la clé publique sur notre machine Debian :

[centreon-engine@localhost ~]$ ssh-copy-id debian@192.168.7.1

/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/var/lib/centreon-engine/.ssh/id\_rsa.pub"

The authenticity of host '192.168.7.1 (192.168.7.1)' can't be established.

ECDSA key fingerprint is SHA256:BcoDPQatqsy2M4roUoiKCrU4nDeJd4cybyaqVkCfWZQ.

Are you sure you want to continue connecting (yes/no/[fingerprint])? yes

/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed

/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys

+-+-+-+-+-+-+-+-+-+-+-+-+-++-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-++-+-+-+

LINUXVMIMAGES.COM

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

User Name: debian

Password: debian (sudo su -)

debian@192.168.7.1's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'debian@192.168.7.1'"

and check to make sure that only the key(s) you wanted were added.

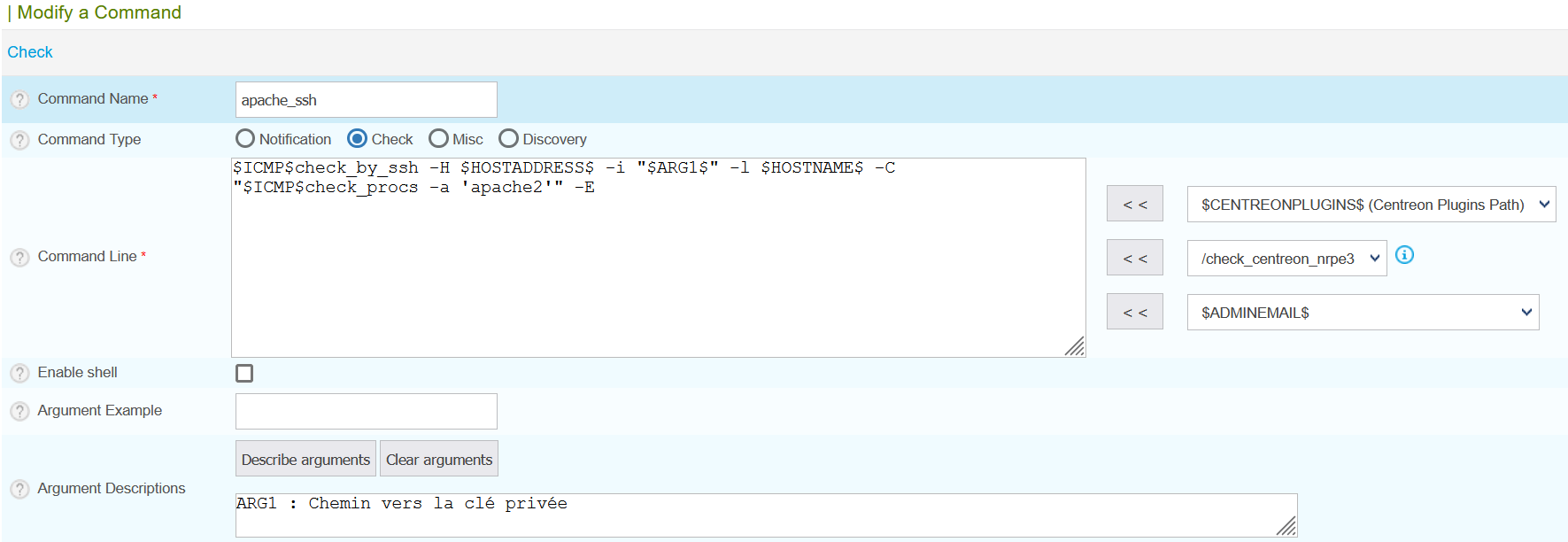
**Check By SSH Apache :**

A présent nous pouvons tester une commande de check by ssh sur le service Apache2 :

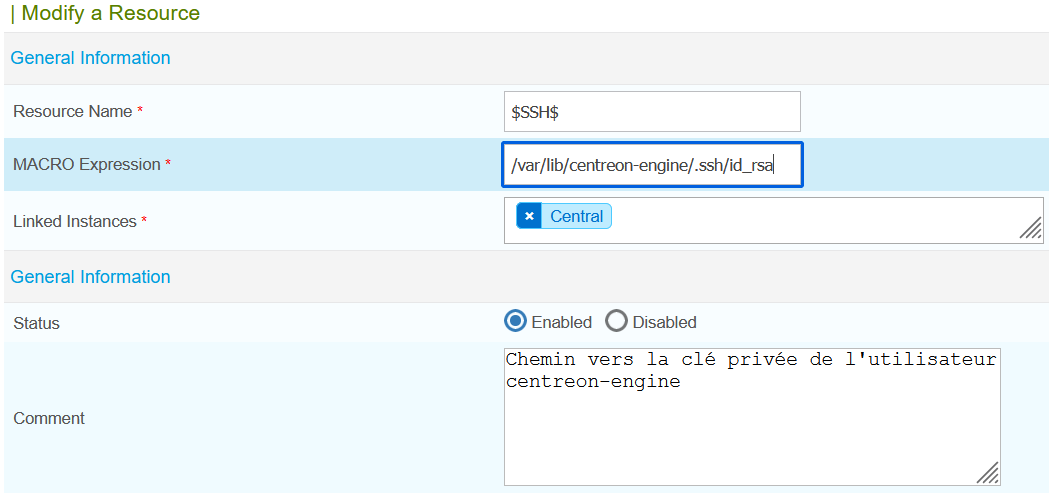
[centreon-engine@localhost ~]$ /usr/local/nagios/libexec/check\_by\_ssh -H 192.168.7.1 -i "/var/lib/centreon-engine/.ssh/id\_rsa" -l debian -C "/usr/local/nagios/libexec/check\_procs -a 'apache2'" -E

PROCS OK: 6 processes with args 'apache2' | procs=6;;;0;

Création de la commande sur centreon :



Au lieu de mettre $ARG1$ pour remplacer le chemin de la clé privée nous aurions pu créer une ressource comme ci-dessous et mettre $SSH$ dans la commande à la place de $ARG1$ :

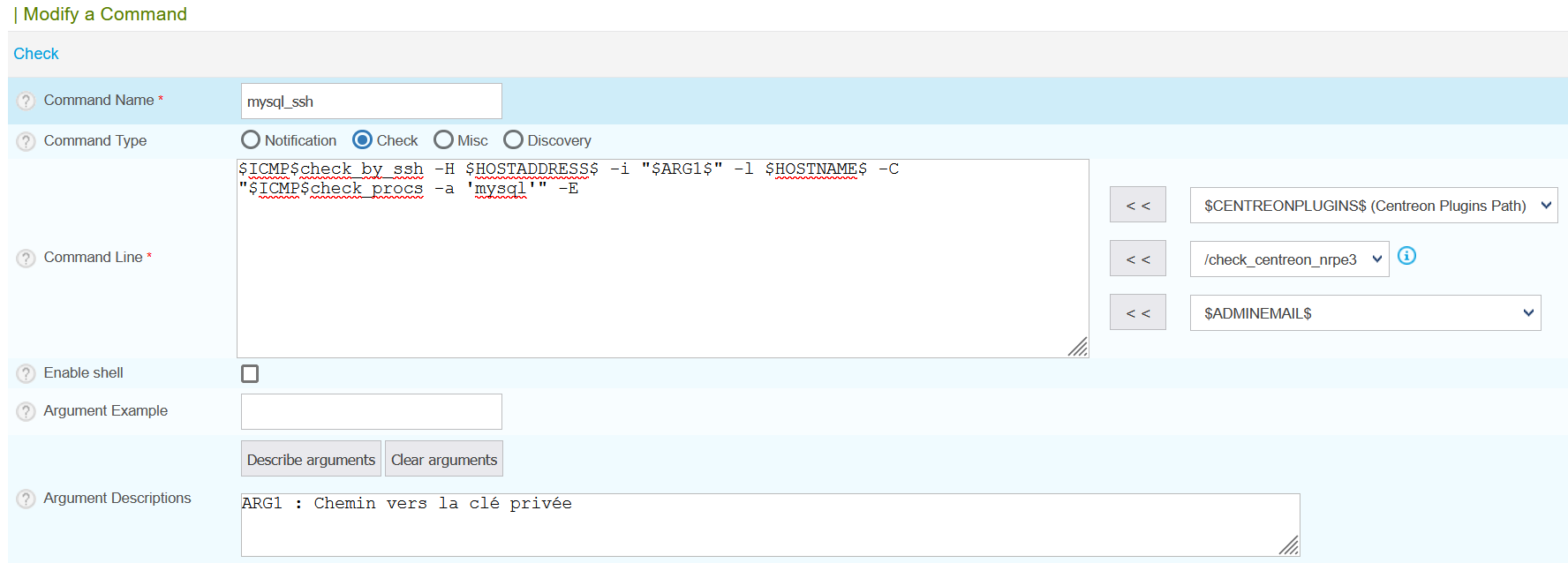


**Check By SSH Mysql :**

[centreon-engine@localhost ~]$ /usr/local/nagios/libexec/check\_by\_ssh -H 192.168.7.1 -i "/var/lib/centreon-engine/.ssh/id\_rsa" -l debian -C "/usr/local/nagios/libexec/check\_procs -a 'mysql'" -E

PROCS OK: 1 process with args 'mysql' | procs=1;;;0;

Création de la commande sur centreon :

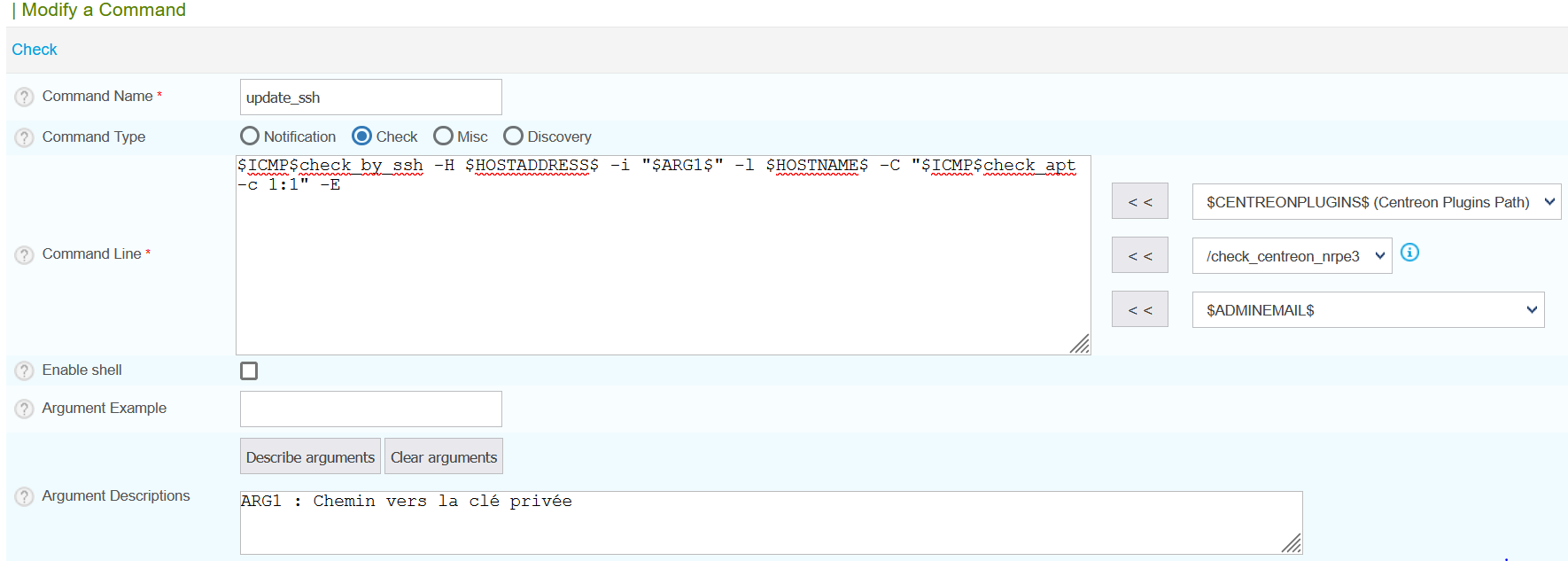


**Check update By SSH :**

[centreon-engine@localhost ~]$ /usr/local/nagios/libexec//check\_by\_ssh -H 192.168.7.1 -i "/var/lib/centreon-engine/.ssh/id\_rsa" -l debian -C "/usr/local/nagios/libexec/check\_apt -c 1:1" -E

APT WARNING: 35 packages available for upgrade (0 critical updates). |available\_upgrades=35;;;0 critical\_updates=0;;;0

Création de la commande sur centreon :



**Machine Windows Server :**

**Vérification de la charge CPU via SNMP avec taux de warning à 80% et taux de criticité à 90% :**

Ligne de commande :

/usr/lib/centreon/plugins//centreon\_windows\_snmp.pl --plugin=os::windows::snmp::plugin --mode=cpu --hostname=192.168.6.1 --snmp-community=public --snmp-version=2 --warning-core=80 --critical-core=90 --verbose

Création de la commande sur centreon :

**Une image contenant texte

Description générée automatiquement**

**Vérification de la RAM via SNMP avec taux de warning à 80% et taux de criticité à 90% :**

Ligne de commande :

/usr/lib/centreon/plugins//centreon\_windows\_snmp.pl --plugin=os::windows::snmp::plugin --mode=memory --hostname=192.168.6.1 --snmp-community=public --snmp-version=2 --warning-usage=80 --critical-usage=90

Création de la commande sur centreon :

Une image contenant texte

Description générée automatiquement

**Vérification du stockage sur le disque C via SNMP avec taux de warning à 80% et taux de criticité à 90% :**

Ligne de commande :

/usr/lib/centreon/plugins//centreon\_windows\_snmp.pl --plugin=os::windows::snmp::plugin --mode=storage --name="C:" --hostname=192.168.6.1 --snmp-community=public --snmp-version=2 --warning-usage=80 --critical-usage=90

Création de la commande sur centreon :

Une image contenant texte

Description générée automatiquement

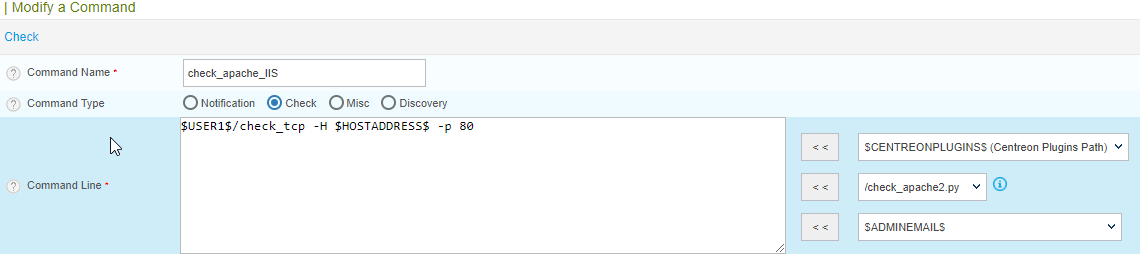
**Vérification du bon fonctionnement de IIS :**

Le but est, comme pour le check vers Apache, de tenter une connexion sur le port 80 afin de voir si il est ouvert ou fermé, ce qui nous indique si IIS ou Apache fonctionne correctement.

Ligne de commande :

/usr/lib64/nagios/plugins/check\_tcp -H 192.168.6.1 -p 80

Création de la commande sur centreon :



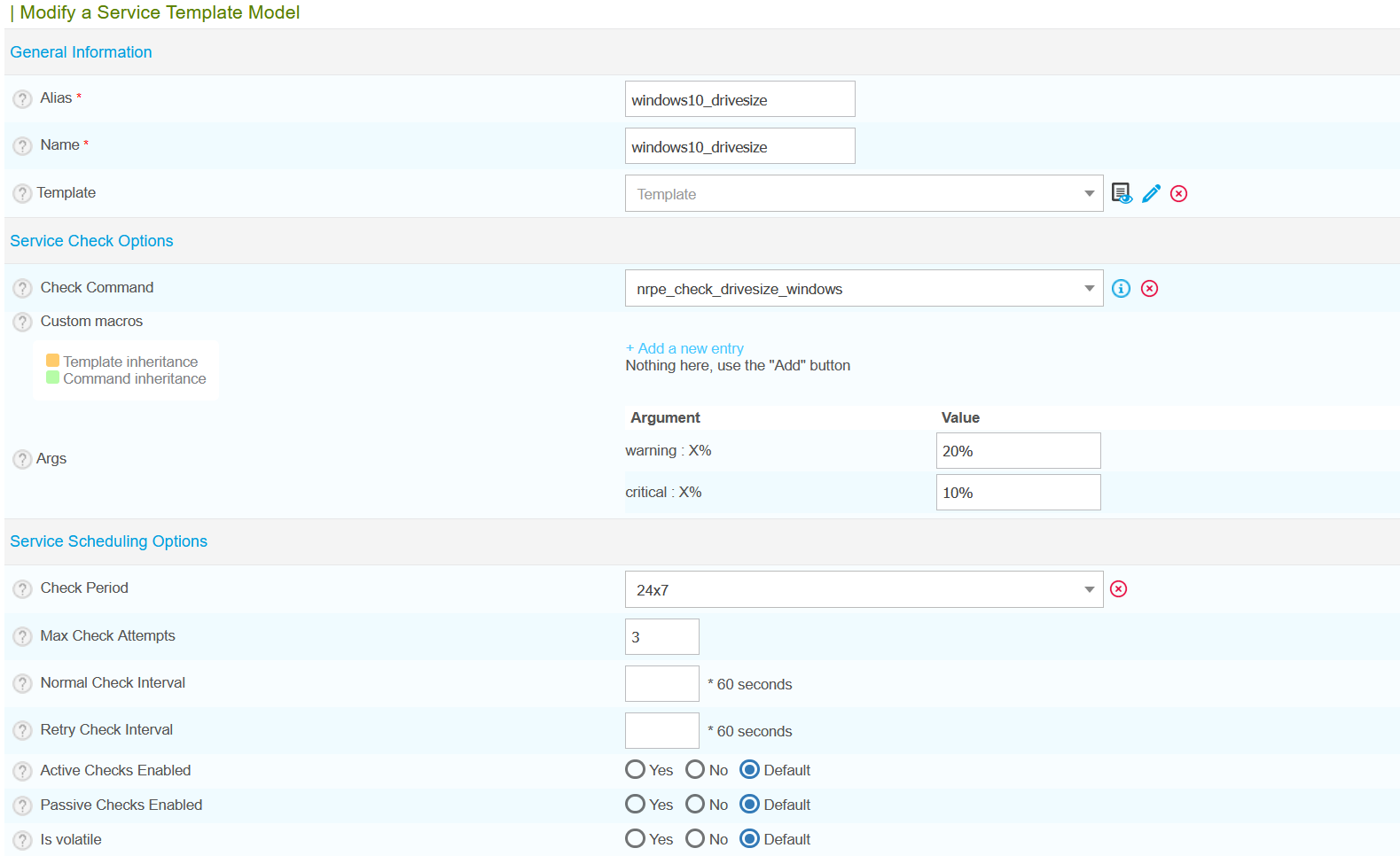
# 2. Création des templates de services

Maintenant que les commandes sont créées dans centreon il faut créer des templates de services qui seront liés à ces commandes (dans l’onglet Configuration -> Services -> Templates) :

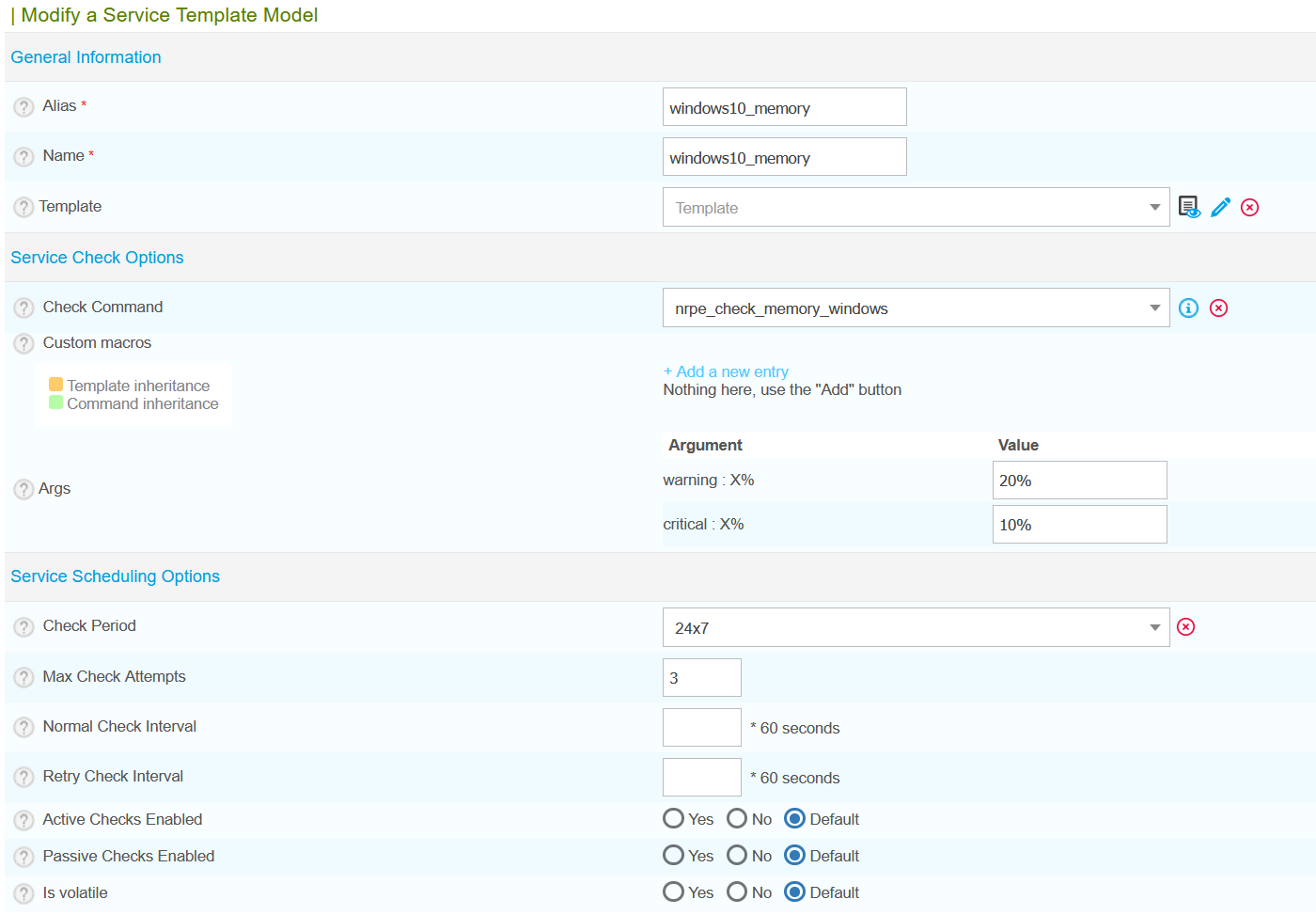
**/ ! \ Il faut faire attention à bien donner les arguments qui vont remplacer les variables dans la commande (voir image ci-dessous)**

**Machine Windows 10 :**

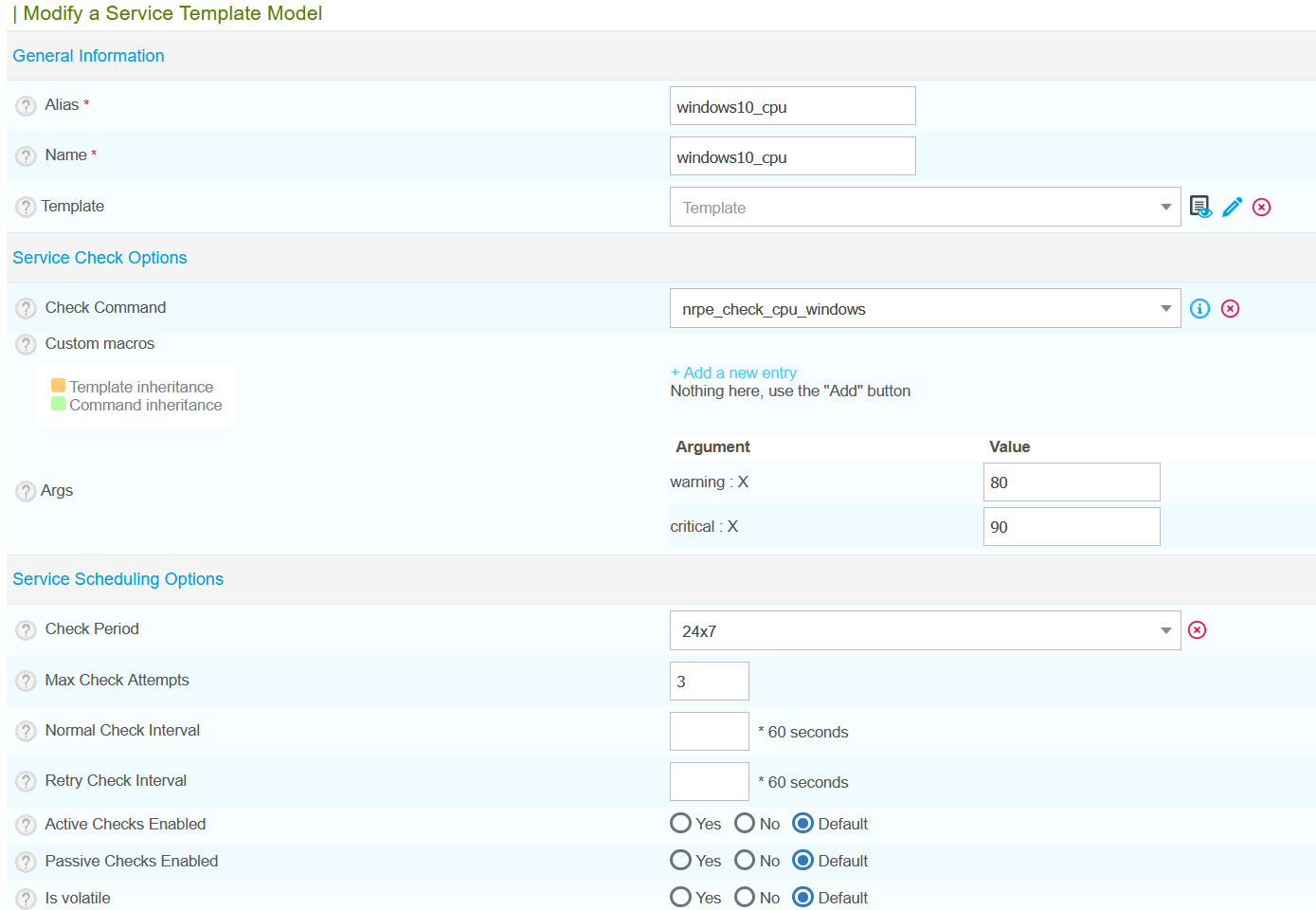
**Vérification du stockage sur le disque C en commande NRPE :**



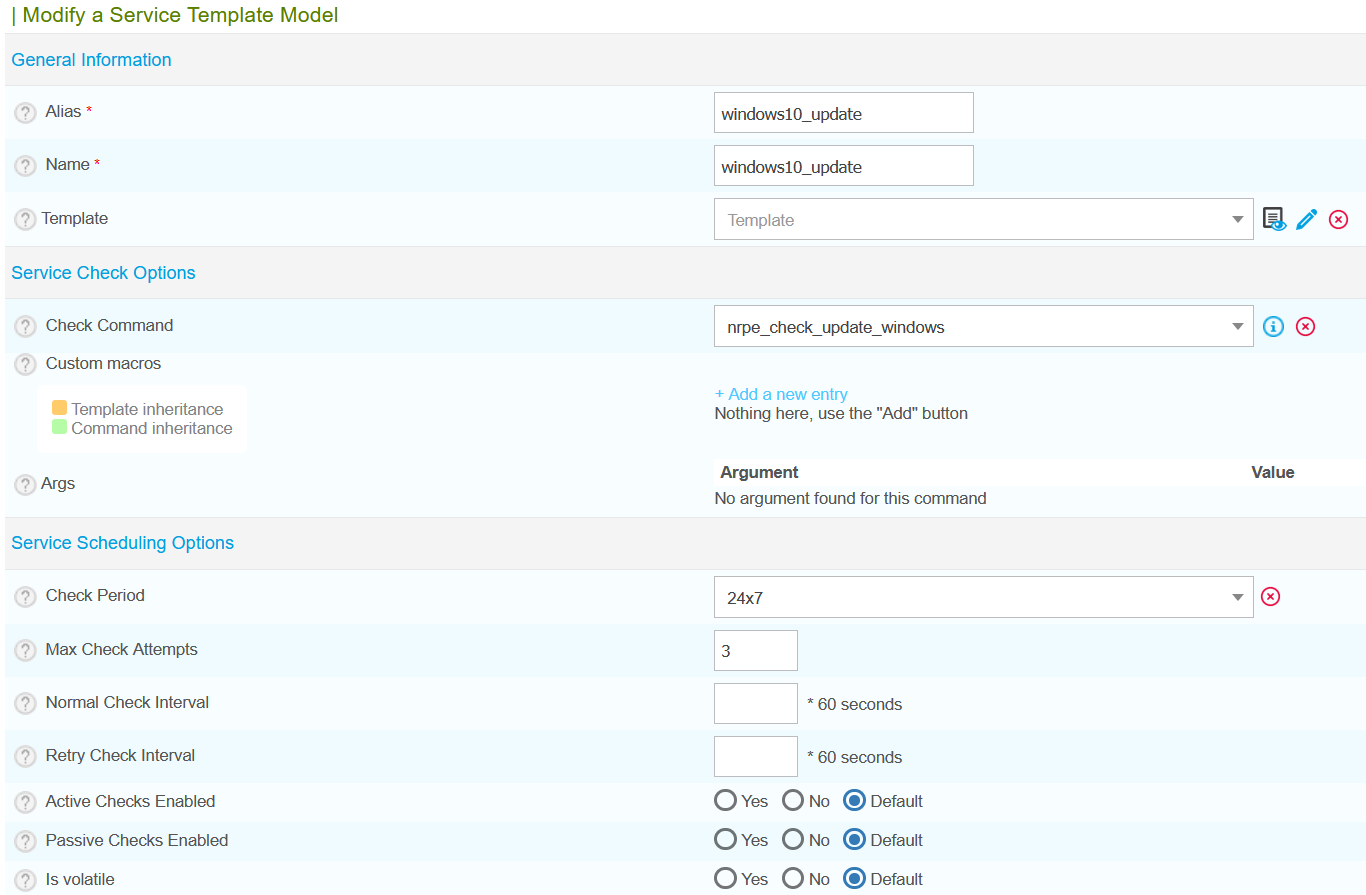
**Vérification de la quantité de mémoire en commande NRPE :**



**Vérification de la charge CPU en commande NRPE :**



**Vérification des mises à jour en commande NRPE :**



**Machine Debian :**

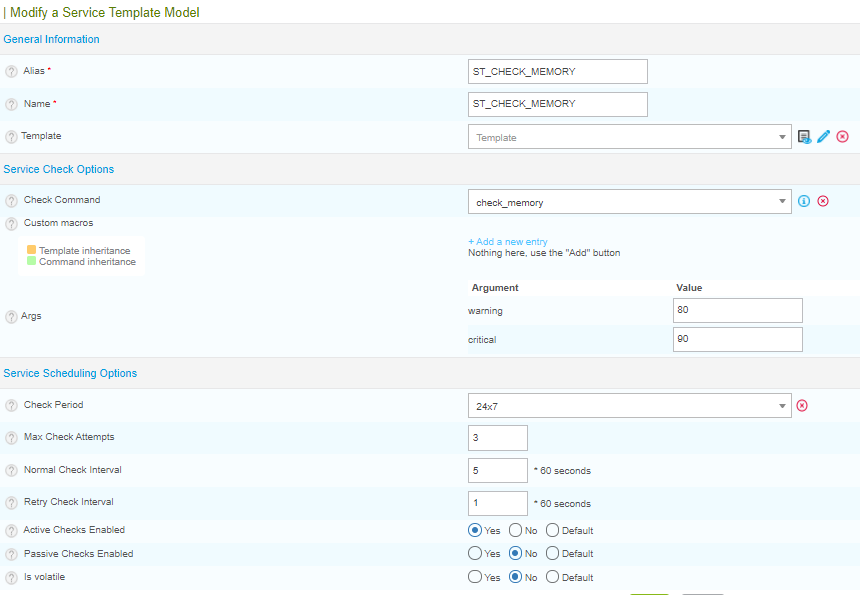
**Partie SNMP :**

**Vérification de la charge CPU :**

Une image contenant table

Description générée automatiquement

**Vérification de la RAM :**

****

**Vérification du stockage :**

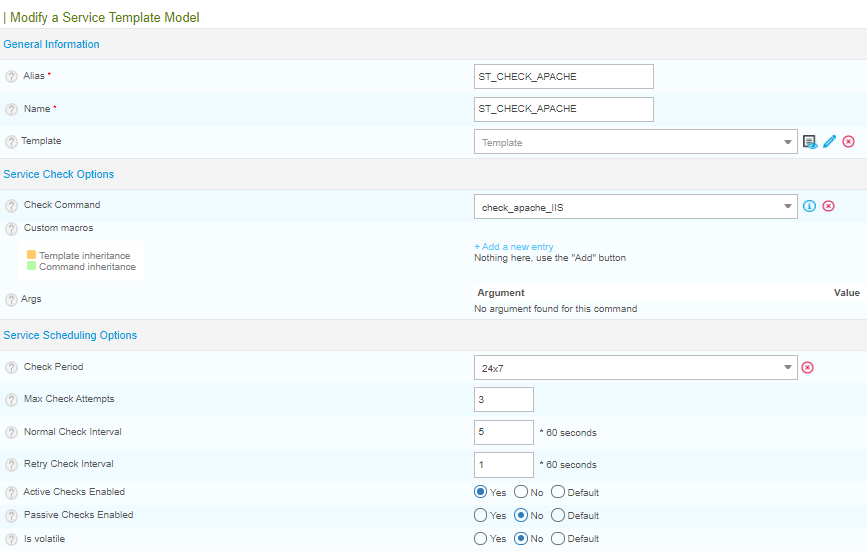
**Une image contenant table

Description générée automatiquement**

**Une image contenant table

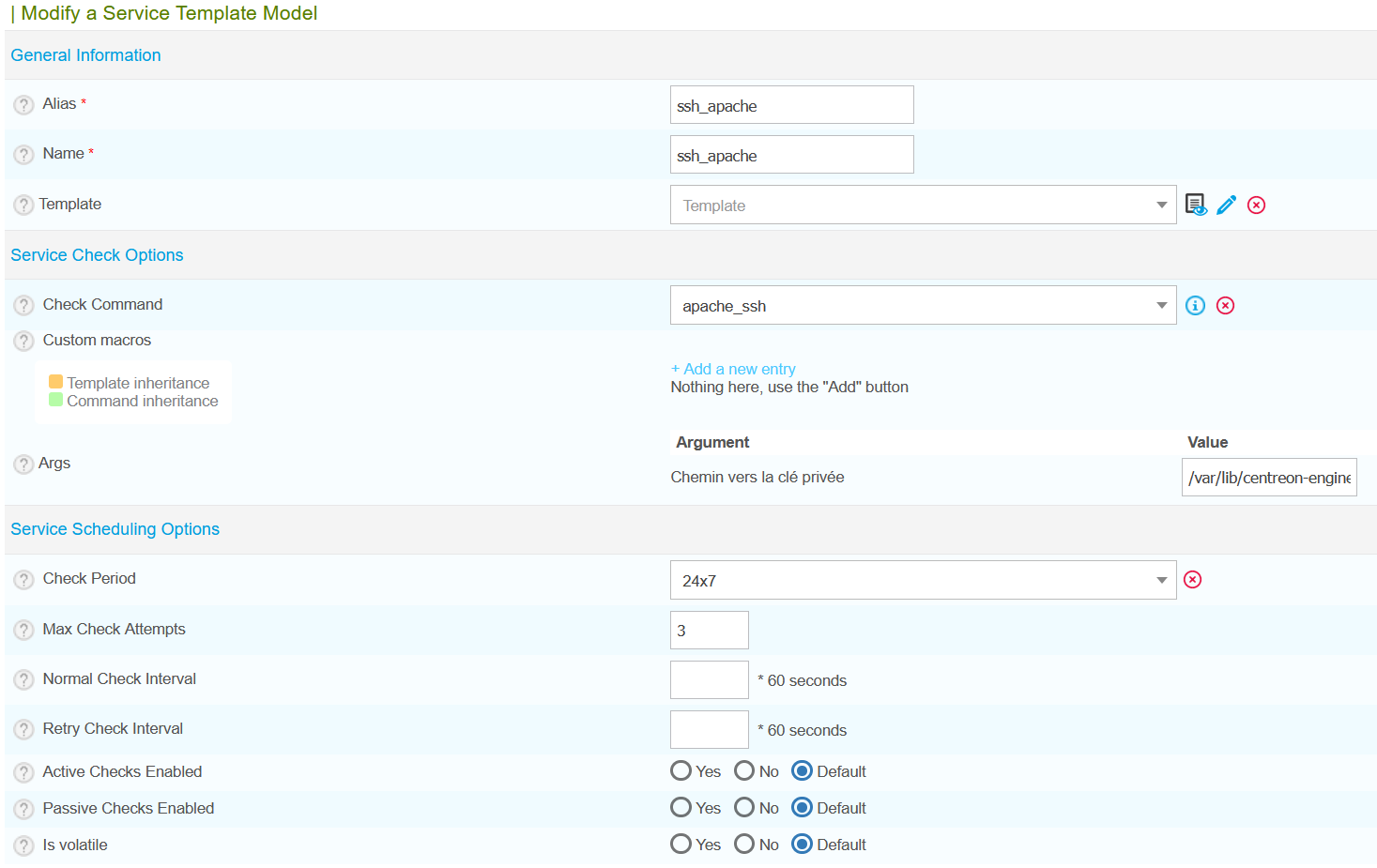
Description générée automatiquement**

**Vérification du fonctionnement d’Apache :**

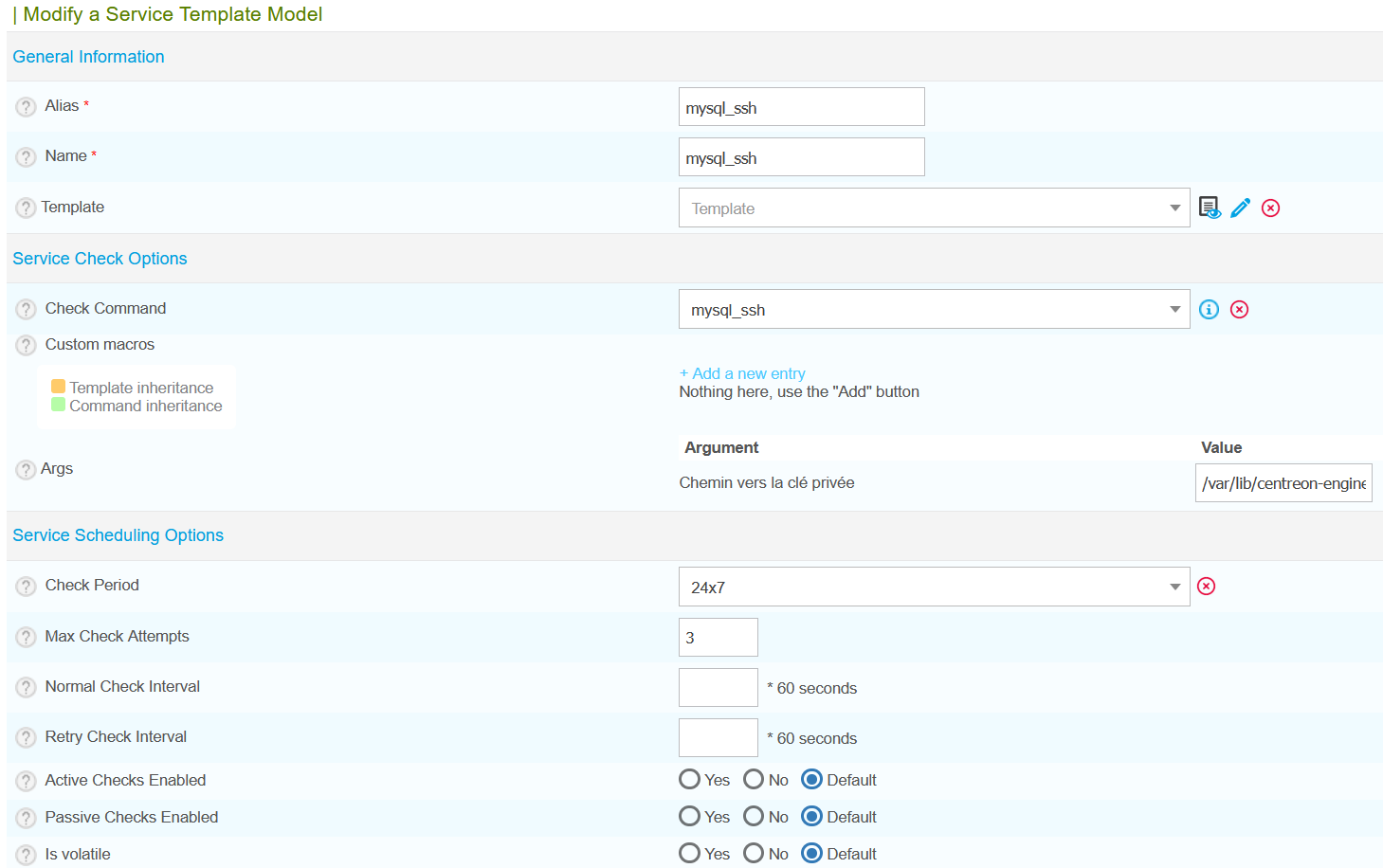
****

**Partie SSH :**

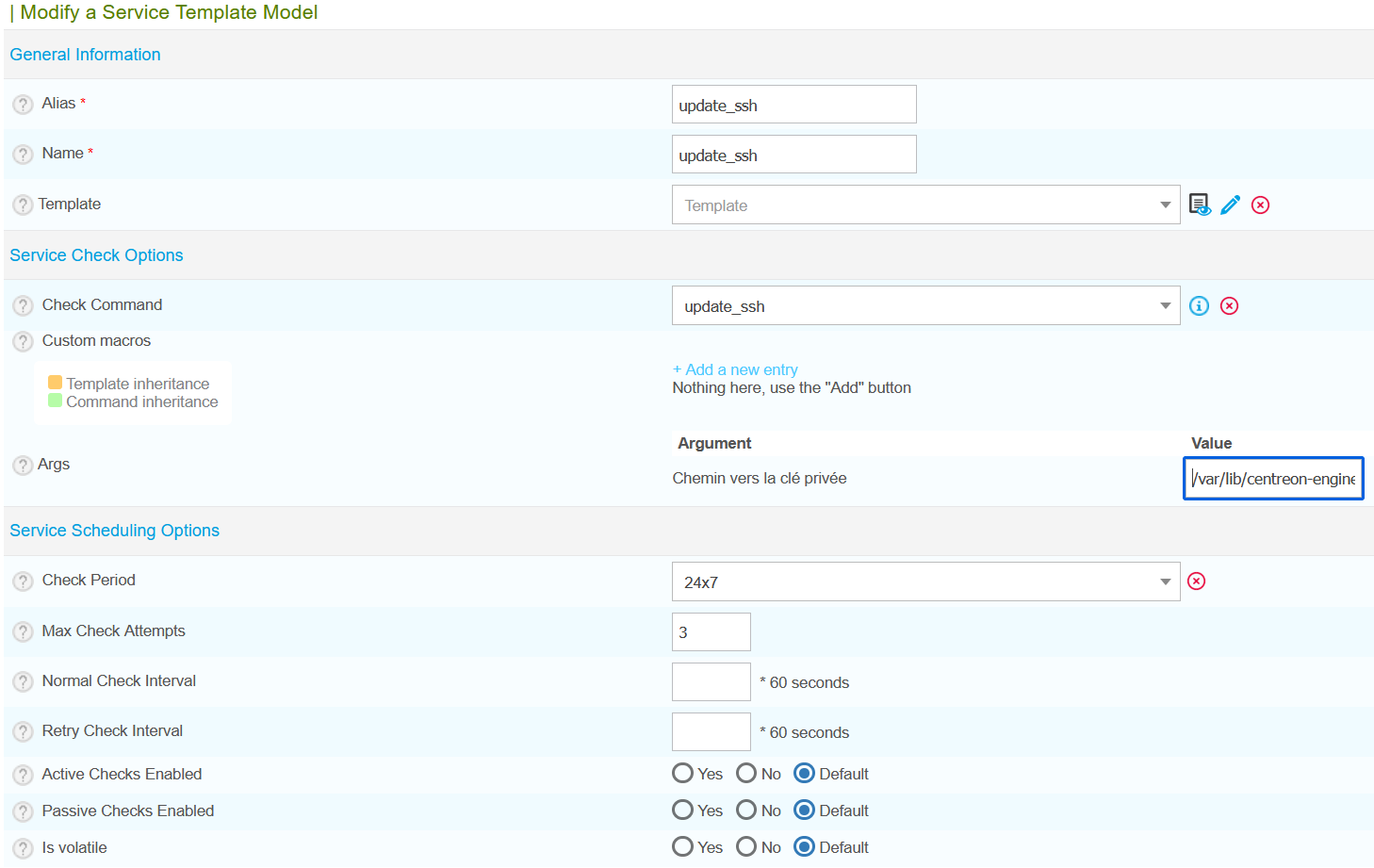
**Vérification d’apache :**



**Vérification de MySQL :**



**Vérification des mises à jour :**



**Machine Windows Server :**

**Vérification du CPU :**

**Une image contenant table

Description générée automatiquement**

**Vérification de la RAM :**

Une image contenant table

Description générée automatiquement

**Vérification du stockage :**

Une image contenant table

Description générée automatiquement

**Vérification du fonctionnement de IIS :**

Comme dit dans la partie précédente, la vérification du fonctionnement d’Apache et de IIS est très similaire. Le template de service est donc le même tout comme la ligne de commande.

Une image contenant table

Description générée automatiquement

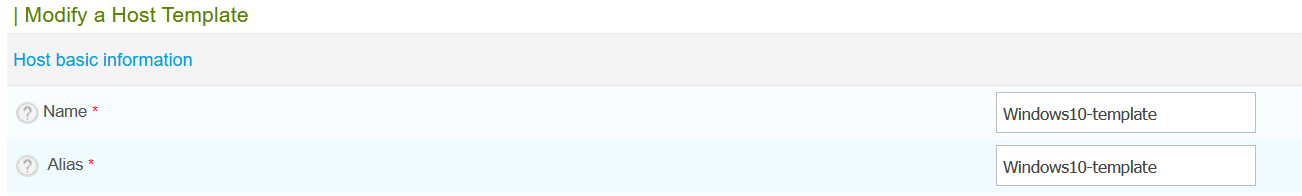
# 3. Création des templates d’hôtes

Une fois les templates de services crées il faut créer des templates d’hôtes pour chaque hôtes (dans l’onglet Configuration -> Hosts -> Templates) :

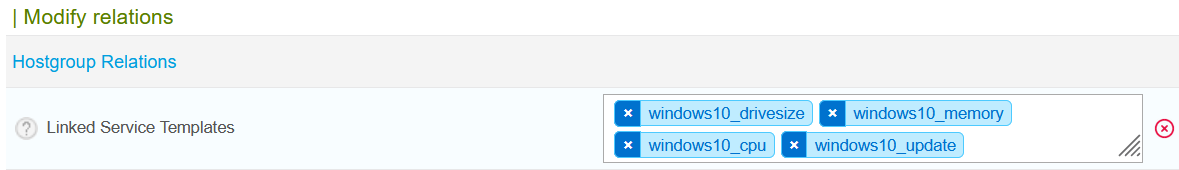
Ici nous devons en faire 3 (un pour chaque machine) :

**Machine Windows 10 :**

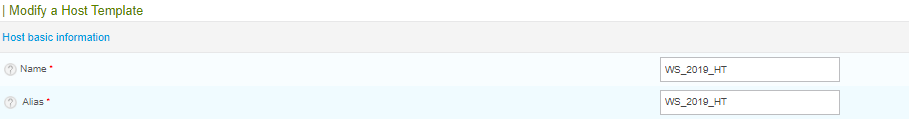
On crée un template pour notre machine Windows 10 avec un nom et un alias :



Dans l’onglet "Relations" on associe les templates de services que l’on veut pour cet hôte :

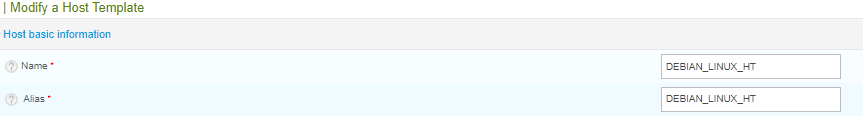


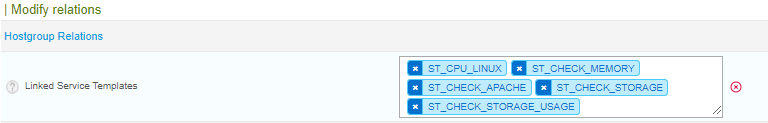
**Machine Windows Server :**

****

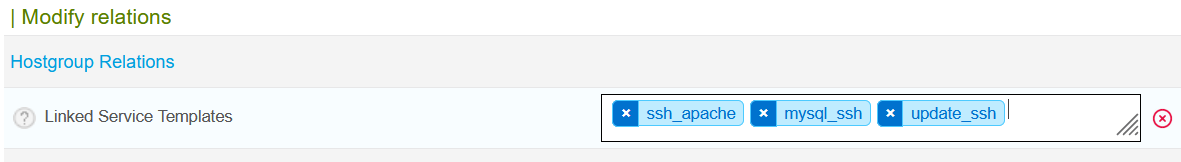
****

**Machine Debian 10 :**

****

****

**Partie SSH Debian 10 :**

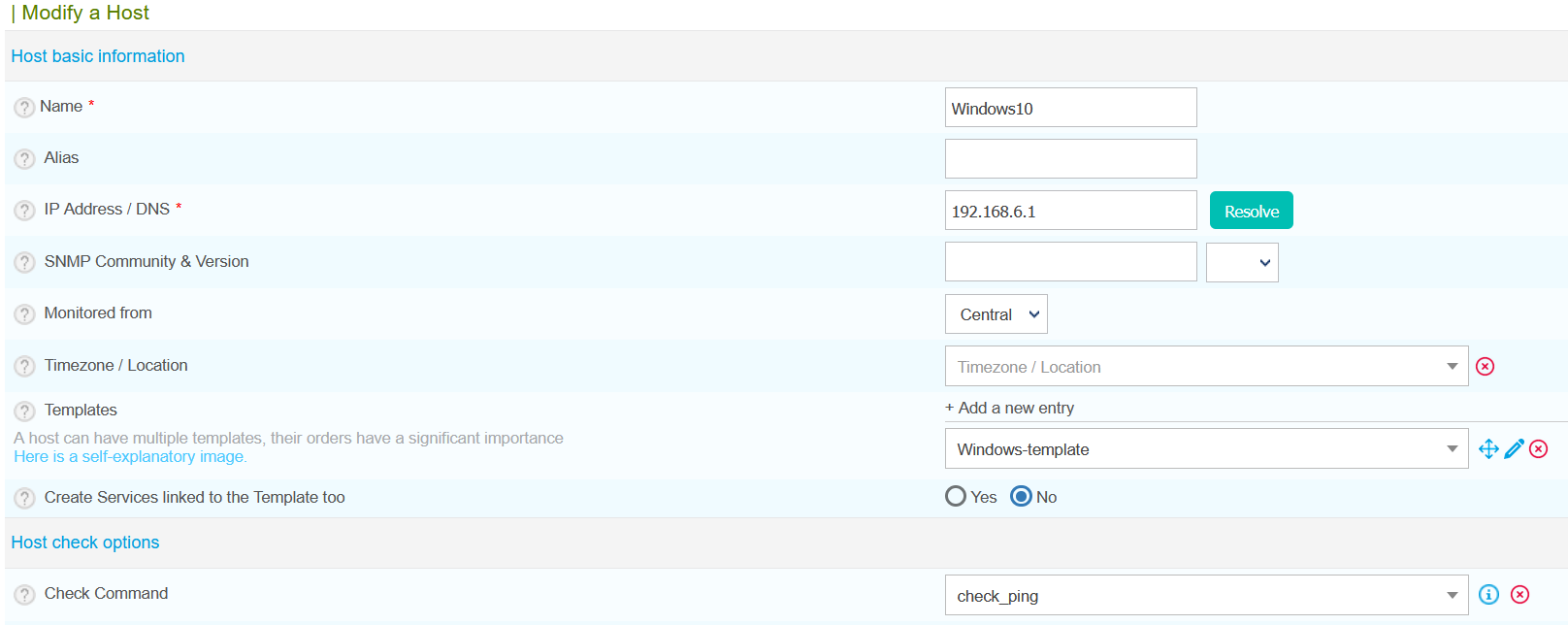


# 4. Ajout des hôtes dans le serveur

Nous devons maintenant ajouter les hôtes dans le serveur (dans l’onglet Configuration -> Hosts -> Hosts) :

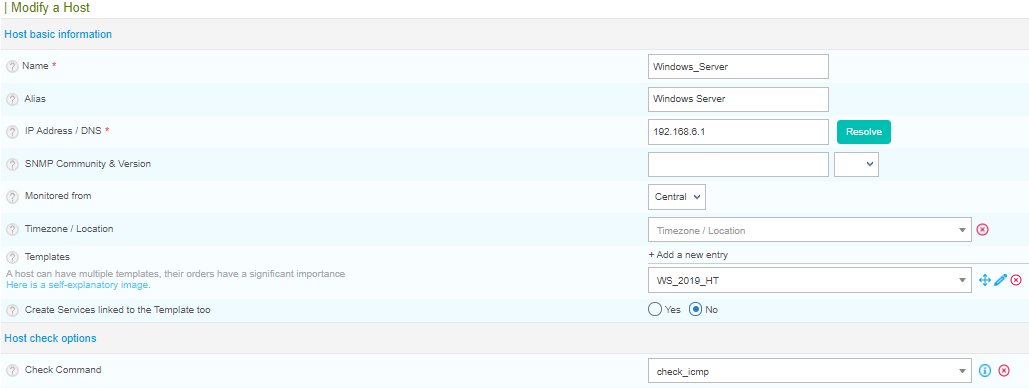
Nous devons définir un nom et une IP pour notre machine Windows 10 et le lier au template d’hôte que nous voulons utiliser :

**Machine Windows 10 :**

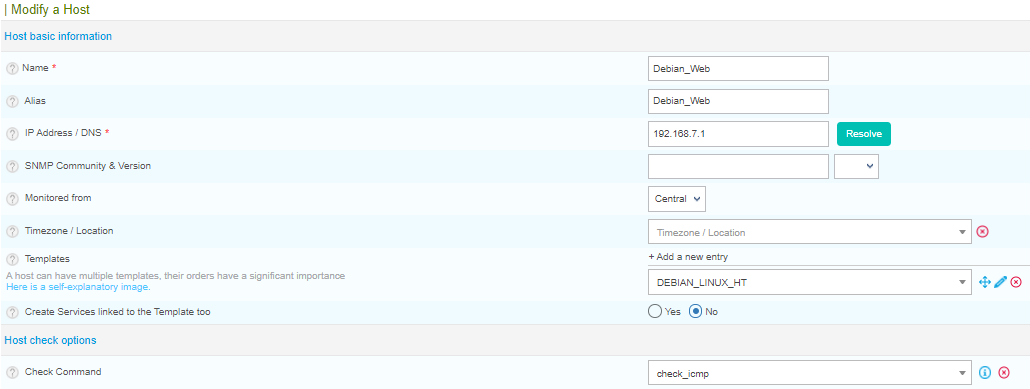


Pour ce qui est du ping il ne faut pas l’intégrer dans le template d’hôte mais le mettre directement sur l’hôte sinon pour la suite dans l’onglet monitoring nous verrons un statut "Pending" sur l’hôte

**Machine Windows Server : (L’IP est la même que Windows 10 car machine de Lucas)**

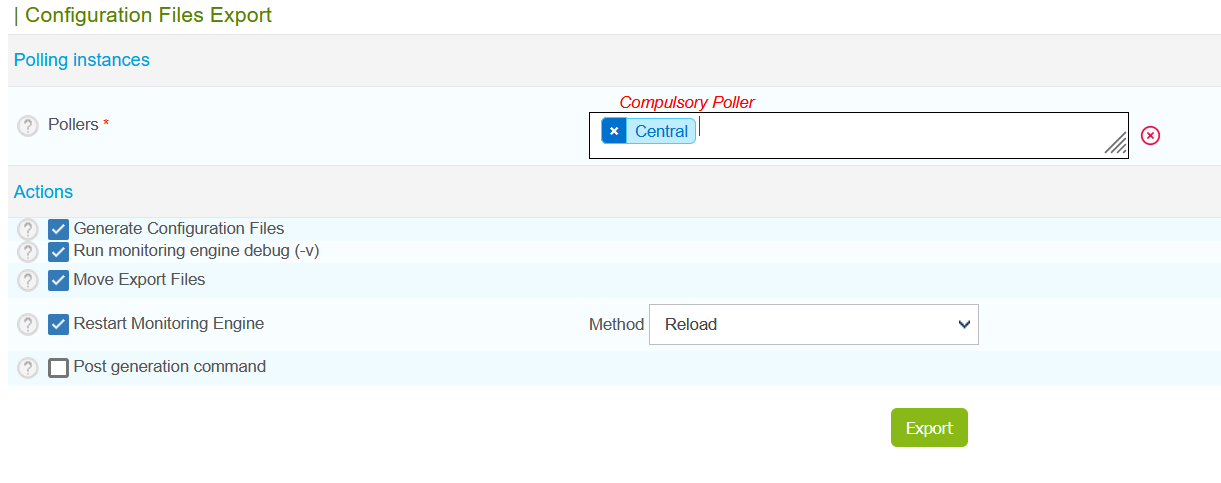
****

**Machine Debian 10 :**

****

# 5. Application des modifications sur le serveur

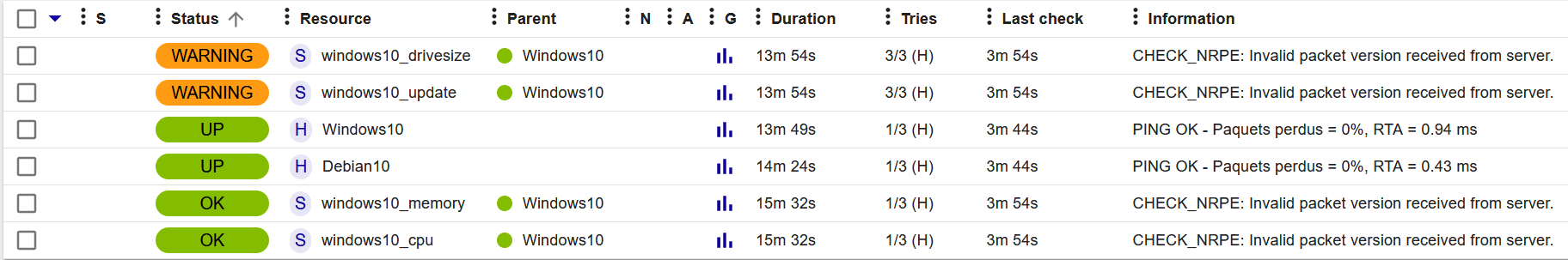
Une fois les modifications apportées sur le serveur il faut relancer sa nouvelle configuration dans l’onglet Pollers -> Pollers -> Export configuration



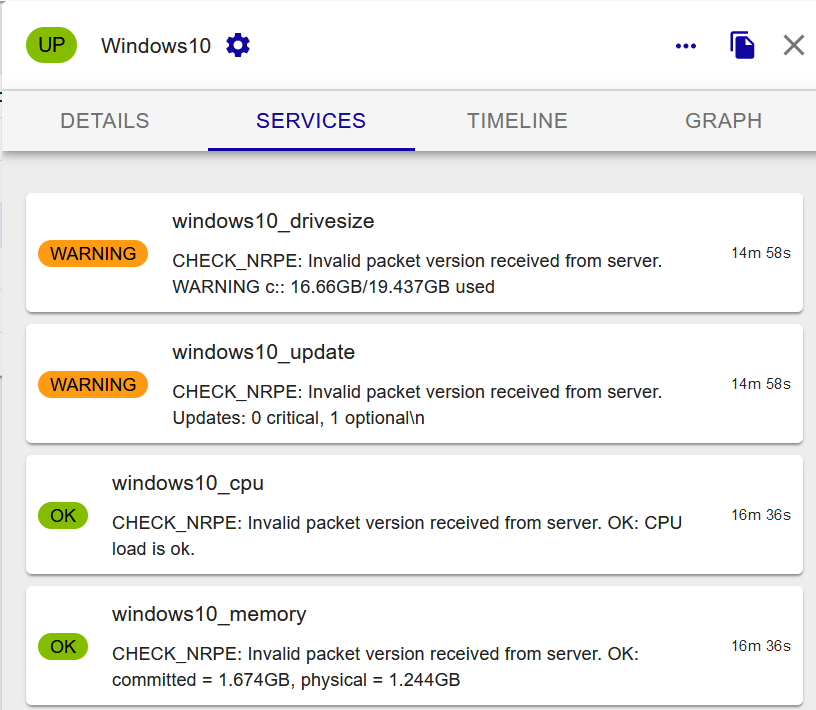
# 6. Monitoring des hôtes

**Machine Windows 10 :**

Nous pouvons maintenant vérifier que les services et les hôtes sont bien supervisé dans notre serveur :



Nous pouvons aussi cliquer sur l’hôte Windows 10 et vérifier ses services :



**Machine Windows Server :**

**Une image contenant texte

Description générée automatiquement**

**Machine Debian 10 :**

**Une image contenant texte

Description générée automatiquement**

**Partie SSH :**

