

TP N°7 BLOC N°2– Bis



Supervision des serveurs et des systèmes avec NAGIOS

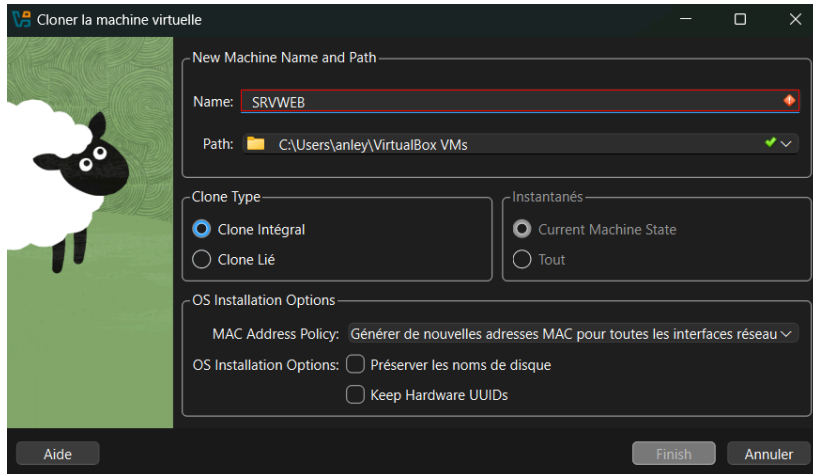
SOMMAIRE

Supervision du serveur « SRVWEB » hébergeant le site de la ville avec « SRVNAGIOS »

Etape 1 : Intégration d'un serveur web Apache 2 hébergeant le site de la ville des Abymes sur « SRVWEB »	Page 1
Etape 2 : Installation et configuration de l'agent "NRPE" et des plugins de Nagios sur le serveur Linux "SRVWEB" à superviser.....	Page 1 à 3
Etape 3 : Installation et configuration de l'agent "NRPE" sur le serveur Nagios "SRVNAGIOS"	Page 3
Etape 4 : Configuration du serveur de supervision "SRVNAGIOS" pour superviser le serveur web distant "SRVWEB"	Page 4 et 5
Etape 5 : Configuration des objets et services à superviser sur le serveur "SRVWEB" par le serveur "SRVNAGIOS"	Page 6

Etape 1 : Intégration d'un serveur web Apache 2 hébergeant le site de la ville des Abymes sur « SRVWEB »

Pour commencer ce TP nous allons commencer par cloner la machine virtuel **SRVWEB0** (APACHE2 INSTALLE) que nous allons appelé **SRVWEB** en clone **intégral** en oubliant pas de **générez de nouvelles adresses MAC pour toutes les interfaces réseau**



Etape 2 : Installation et configuration de l'agent "NRPE" et des plugins de Nagios sur le serveur Linux "SRVWEB" à superviser

Pour commencer l'installation et la configuration de l'agent NRPE sur le serveur web nous allons tout d'abord mettre à niveau le système Linux Debian du serveur en tapant les commande **" apt update " & "apt upgrade"**.

Après avoir mis à jour le système nous allons installer différents paquets pour que l'agent NRPE puisse bien fonctionner. Pour cela installez les différents paquets en saisissant la commande **"pt install build-essential, apache2, php, openssl, libssl-dev, libgd-dev, gcc, make, autoconf, libc6, libapache2-mod-php, wget, apache2-utils et unzip"**.

Maintenant nous allons installer l'archive **nrpe-4.1.0.tar.gz** depuis l'URL : **"https://github.com/NagiosEnterprises/nrpe/archive/nrpe-4.1.0.tar.gz"**. avant l'installation de cette archive, accédez au répertoire **/tmp** (ce sera dans ce répertoire où l'archive sera installée) . Pour cela saisissez la commande **"cd /tmp"**. Maintenant pour installer l'archive tapez la commande **"wget https://github.com/NagiosEnterprises/nrpe/archive/nrpe-4.1.0.tar.gz"**. l'archive **nrpe-4.1.0.tar.gz** a bien été installé

```
root@srvweb:/tmp# ls nrpe-4.1.0.tar.gz
nrpe-4.1.0.tar.gz
```

Puisque ce fichier est compressé nous allons le décompressé en tapant la commande **"tar xvf nrpe-4.1.0.tar.gz"** . Maintenant accédez au répertoire nrpe-nrpe-4.1.0 en tapant la commande **"cd nrpe-nrpe-4.1.0"** . Maintenant nous allons taper certaine commande pour la compilation et l'installions de NRPE:

1. **./configure - -enable-command-args**
2. **make all**
3. **make install-groups-users**

(Si une erreur apparait en saisissant la commande **"make install-groups-users"** , rajoutez devant la commande (sudo))

```
root@srvnagios:/tmp/nrpe-nrpe-4.1.0# sudo make install-groups-users
Group nagios already exists
User nagios already exists
```

***capture après exécution de la commande "sudo make install-groups-users"**

Ici ont constate que le Groupe utilisateur nagios à bien été créé

Terminons la compilation du paquet NRPE en saisant le reste des commandes:

4. **make install**
5. **make install-config**
6. **make install-init**

Après avoir saisi les commandes pour la compilation et l'installation de NRPE sur le serveur SRVWEB, nous allons installer les plugins de NAGIOS toujours sur ce serveur.

Tout comme pour l'installation de NRPE, nous allons procéder presque de la même manière pour l'installation des plugins NAGIOS.

Nous allons accéder au répertoire **/tmp** en saisissant la commande **cd /tmp**. Maintenant nous allons télécharger l'archive **"nagios-plugins-2.4.12.tar.gz"** depuis l'URL **"<https://github.com/nagios-plugins/nagios-plugins/releases/download/release-2.4.12/nagiosplugins-2.4.12.tar.gz>"**. Pour ce faire saisissez la commande **"wget <https://github.com/nagios-plugins/nagios-plugins/releases/download/release-2.4.12/nagiosplugins-2.4.12.tar.gz>"**.

Toujours dans le répertoire, nous allons décompresser le fichier **nagios-plugins-2.4.12.tar.gz** en tapant la commande **"tar xvfz nagios-plugins-2.4.12.tar.gz"**. Maintenant accédons au répertoire **nagios-plugins-2.4.12** en saisissant la commande **"cd /tmp/nagios-plugins-2.4.12"**. Dans ce répertoire nous allons exécuter certaines commandes :

1. **./configure**
2. **make**
3. **make install**

Après la compilation et l'installation de l'agent NRPE et des plugins Nagios, on va vérifier que les fichiers de configuration des plugins de Nagios sont au bon emplacement, pour ce faire

saisissez la commande, **"ls /usr/local/nagios/libexec"**

```
root@srvweb:/usr/local/nagios/libexec# ls /usr/local/nagios/libexec/
check_apr      check_disk      check_http      check_jabber    check_nttp      check_nwstat    check_rpc        check_ssmtp     check_users
check_breeze   check_disk_smb  check_icmp      check_load      check_nntp      check_oracle    check_sensors    check_swap      check_wave
check_by_ssh   check_dns       check_ide_smart  check_log       check_nrpe      check_overcr    check_simap      check_tcp       negate
check_clamd    check_dummy     check_ifoperstatus check_mailq     check_nt        check_ping      check_smtp       check_time      remove_perfdata
check_cluster  check_file_age  check_ifstatus   check_mrtg      check_ntp       check_pop       check_spop       check_udp       urlize
check_dhcp     check_flexlm    check_imap      check_mrtgtraf  check_ntp_peer  check_procs     check_ssh        check_ups       utils.pm
check_dig      check_ftp       check_ircd      check_nagios    check_ntp_time  check_real      check_ssl_validity check_uptime     utils.sh
```

*Capture du contenu du répertoire **/usr/local/nagios/libexec**

Constate que le module **check_nrpe** est bien présent dans le répertoire **/usr/local/nagios/libexec**

et en saisissant la commande **"ls /usr/local/nagios/etc/"** on constate que le fichier de configuration **nrpe.cfg** est bien présent

```
root@srvweb:/usr/local/nagios/libexec# ls /usr/local/nagios/etc/
nrpe.cfg  nrpe.cfg.save
```

*Capture réalisée après certaine étape c'est la raison pour laquelle **nrpe.cfg.save** est déjà là

Après avoir vérifier le contenu des différents répertoires nous allons réaliser une copie de la configuration du fichier `nrpe.cfg` en `nrpe.cfg.save`. Pour cela saisissez la commande **`cp /usr/local/nagios/etc/nrpe.cfg /usr/local/nagios/etc/nrpe.cfg.save`** .

Maintenant éditons le fichier `nrpe.cfg` avec la commande **`nano /usr/local/nagios/etc/nrpe.cfg`**

```
allowed_hosts=127.0.0.1,192.168.4.5

# 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=1
```

ajouter l'adresse IP de votre serveur Nagios dans mon cas ce sera **192.168.4.5** .

et également dans la partie `"dont_blame_nrpe="` remplacer le 0 par 1

la ligne : `"allowed_hosts = 127.0.0.1, adresse_IP_serveur_Nagios`

`127.0.0.1` autorise les tests en local (avec le serveur lui-même).

`adresse_IP_serveur_Nagios (192.168.45.5)` autorise le serveur Nagios à interroger le
Serveur web sur sont été

`dont_blame_nrpe=1` permet à Nagios de passer des arguments aux commandes NRPE

Après toutes ses installation et configuration nous allons relancez et activez le service nrpe:

1. `systemctl enable nrpe.service`
2. `systemctl start nrpe.service`
3. `systemctl restart nrpe.service`

Maintenant testons le module `check_nrpe`, pour cela a déplacez vous dans le répertoire `/usr/local/nagios/libexec` en saisissant la commande `"cd /usr/local/nagios/libexec"`. Saisissez la commande `./check_nrpe -H 127.0.0.1`

```
root@srvweb:/usr/local/nagios/libexec# ./check_nrpe -H 127.0.0.1
NRPE v4.1.0
root@srvweb:/usr/local/nagios/libexec# _
```

Ont constate que cette indique la version de NRPE installer (NRPE V4.1.0)

Etape 3 : Installation et configuration de l'agent "NRPE" sur le serveur Nagios "SRVNAGIOS"

Après avoir tous installer sur le serveur WEB, nous allons basculer sur le serveur Nagios. Qu'allons nous faire ?

nous également comme pour le Serveur Web télécharger l'agent NRPE réaliser quasiment presque les même étape installions tout d'abords les paquets nécessaire pour cela faite `"apt install build-essential, apache2, php, openssl, libssl-dev, libgd-dev, gcc, make, autoconf, libc6, libapache2-mod-php, wget, apache2-utils et unzip -y"`. Après avoir installer les paquets accédez au répertoire `/tmp` en saisissant la commande `"cd /tmp"`. Maintenant télécharger l'archive `nrpe-4.1.0.tar.gz` en saisissant la commande `"wget https://github.com/NagiosEnterprises/nrpe/archive/nrpe-4.1.0.tar.gz"`. Maintenant décompressé le l'archive `nrpe-4.1.0.tar.gz` en saisissant la commande `"tar xvfz nrpe-4.1.0.tar.gz"`. Maintenant déplacez vous dans le répertoire `/tmp` en saisissant la commande `"cd /tmp/nrpe-4.1.0"`. Maintenant tapez les commande pour la compilation :

1. `./configure - --enable-command-args`
2. `make all`
3. `make install-groups-users` (*Si la commande ne passe pas ajouter sudo)
4. `make install`
5. `make install-config`
6. `make install-init`

Testons le module `check_nrpe`. Déplacez vous dans le répertoire `/usr/local/nagios/libexec` en saisissant la commande `"cd /usr/local/nagios/libexec"` et maintenant saisissez la

commande **`“./check_nrpe -H 127.0.0.1”`** pour tester le fonctionnement du module check_nrpe.

```
root@srvnagios:/usr/local/nagios/libexec# ./check_nrpe -H 127.0.0.1
NRPE v4.1.0
```

***Capture depuis le serveur Nagios**

Ici on constate que cette commande affiche la version du NRPE **(SUR LE SERVEUR LUI MEME)** .

Maintenant testons la connexion au serveur web toujours avec le plugin check_nrpe. Pour ce faire saisissez la commande **`“./check_nrpe -H adresse_IP_SRVWEB”`** . Dans mon cas l'adresse ip de mon serveur web 192.168.4.4

```
root@srvnagios:/usr/local/nagios/libexec# ./check_nrpe -H 192.168.4.4
NRPE v4.1.0
```

***Capture depuis le serveur Nagios avec la commande au préalable**

Ici on constate que cette commande affiche la version NRPE installée sur le serveur qui est supervisé (version NRPE démon serveur Web en 192.168.4.4)

Etape 4 : Configuration du serveur de supervision "SRVNAGIOS" pour superviser le serveur web distant "SRVWEB"

Allons sur le serveur nagios et configurons Nagios pour superviser les différent objet sur le serveur Web. Pour ce faire éditons le fichier

"/usr/local/nagios/etc/objects/commands.cfg" via la commande **"nano /usr/local/nagios/etc/objects/commands.cfg"** et ajouter cette ligne de commande après la partie **check_nt**, comme indiquer sur la capture d'écran si dessous

```
define command {  
    command_name    check_nrpe  
    command_line    $USER1$/check_nrpe -H $HOSTADDRESS$ -u -c $ARG1$ -a $ARG2$ $ARG3$ $ARG4$ $$ARG5$ $ARG6$  
}
```

***Capture d'écran du contenu du fichier commands.cfg**

cette ligne permet à Nagios de communiquer avec les serveur distants via NRPE afin d'exécuter des vérifications, sans cette commande Nagios ne peut pas interroger les machines distants (exemple envoyer des requête au serveur distant) .

1. **command_name check_nrpe** : sera le nom de la commande pour interroger le serveur web et savoir son état
2. **command_line** ligne : ligne de commande réelle qui sera exécutée
3. **\$USER1\$** : Macro contenant le chemin vers le répertoire des plugins
4. **-H \$HOSTADDRESS\$** : l'adresse IP de l'hôte distant
5. **-u** : autorise le plugin **check_nrpe** à accepter des arguments non chiffrés (nécessaire si comme précédemment **dont_blame_nrpe=1**)
6. **-a \$ARG2\$...** : Arguments supplémentaires . à passer à la commande distante.

Maintenant créons le répertoire appelé **servers** à l'emplacement **/usr/local/nagios/etc** pour ce faire saisissez la commande **"mkdir /usr/local/nagios/etc/servers"**. Les fichiers de configuration seront là. Maintenant créons le fichier **srvweb.cfg** qui sera dans le répertoire nommé précédemment pour cela faites taper seulement la commande **"nano /usr/local/nagios/etc/servers/srvweb.cfg"**.

Nous allons ajouter quelques lignes de commande dans ce fichier.

```
#Définition de la machine Linux "SRVWEB"  
  
define host {  
    use                linux-server  
    host_name          srvweb  
    alias              Service web Apache2  
    address             192.168.4.4  
}
```

***Capture du contenu du fichier srvweb.cfg**

1. **define host** permet de déclarer un hôte ou un serveur à superviser dans Nagios
2. **use** permet d'indiquer le modèle (template) en fait il applique une configuration prédéfinie et évite de tout écrire manuellement dans mon cas ce sera **linux-server**
3. **host_name** permet d'indiquer le nom unique de l'hôte. en fait c'est l'identifiant utilisé pour lier les services à cet hôte, dans mon cas c'est **srvweb**
4. **alias** permet simplement de décrire l'hôte (quel service par exemple ou quel type de serveur)
5. **address** c'est tout simplement pour indiquer l'adresse IP du serveur supervisé ici l'adresse de mon serveur supervisé est 192.168.4.4

Maintenant éditons le fichier `/usr/local/nagios/etc/templates.cfg` et observons le:

1. **max_check_attempts** : indique le nombre de tentative de vérification avant de considérer l'hôte ou le service comme down ou critical. cela émet les alertes pour des erreurs passagères
2. **check_interval** : indique l'intervalle (en minutes) entre deux vérifications programmées lorsque l'hôte ou le service est dans un état OK
3. **retry_interval**: indique l'intervalle (en minutes) entre les vérifications lorsque l'hôte ou le service est dans un état non-OK
4. **check_period 24x7** : indique la période de temps pendant laquelle les vérifications programmées sont effectuées (ici c'est 24 heures sur 24 et 7 jours sur 7)
5. **check_freshness** : permet de vérifier que le service ne reste pas sans nouvelle donnée trop longtemps. si rien n'est reçu Nagios force un contrôle pour s'assurer que tout va bien
6. **contact_groups** : indique le groupe de contacts qui recevra les notifications d'alerte pour les hôte ou service en cas de problème
7. **notification_interval** : indique (en minutes) auquel une notification est renvoyée si le problème persiste
8. **notification_period 24x7** : indique la période pendant laquelle les notifications peuvent être envoyées
9. **notification_enabled** : Active l'envoi de notifications pour cet objet
10. **notification_options d,u,r** : pour les hôtes, cela spécifie pour quels états envoyer une notification (d=down, u=unreachable, r=recovered). et pour un service c'est w,c,r (w=warning, c=critical, r=recovered).
11. **register** : Indique que cette définition est un objet réel qui doit être enregistré.
12. **active_checks_enabled** : Nagios vérifie lui-même l'état donc des écouteurs actifs
13. **passive_checks_enabled** : C'est le serveur web qui envoie les informations (alerte) directement au cas de panne.

Après avoir expliqué la signification des lignes dans le contenu du fichier `template.cfg` nous paramétrons le fichier de configuration `"/usr/local/nagios/etc/nagios.cfg"` pour cela saisissez la commande `"nano /usr/local/nagios/etc/nagios.cfg"`.

```
cfg_dir=/usr/local/nagios/etc/servers
#cfg_dir=/usr/local/nagios/etc/printers
#cfg_dir=/usr/local/nagios/etc/switches
#cfg_dir=/usr/local/nagios/etc/routers
```

***Capture du documents provenant du contenue du fichier nagios.cfg**

décommentez la ligne : `cfg_dir=/usr/local/nagios/etc/servers`

Maintenant nous allons indiquer le nom du propriétaire et du groupe ou il appartient le répertoire **servers** et **srvweb.cfg**

Pour ce faire saisissez la commande **"ls -a -l /usr/local/nagios/etc/servers"**. vous le nom du propriétaire et le groupe auxquels il appartient

```
root@srvnagios:/usr/local/nagios/libexec# ls -a -l /usr/local/nagios/etc/servers/
total 12
drwxr-xr-x 2 nagios nagios 4096 27 mars 08:13 .
drwxrwxr-x 4 nagios nagios 4096 27 mars 19:28 ..
-rw-r--r-- 1 nagios nagios 1934 26 mars 17:26 srvweb.cfg
```

*Capture réalisé après certaine étape c'est pour quoi le du nom du propriétaire et de son groupe il appartient après

le propriétaire de base de normalement c'était root mais j'ai fait les modification pour que le propriétaire sois nagios voici la commande **"chown -R nagios:nagios /usr/local/nagios/etc/servers"**.

Maintenant vérifions que les paramètres et configurations sont valides, pour ce faire saisissez la commande : **"/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg"**.

```
root@srvnagios:/usr/local/nagios/libexec# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
Nagios Core 4.5.11
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2026-01-14
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 15 services.
  Checked 2 hosts.
  Checked 1 host groups.
  Checked 0 service groups.
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 25 commands.
  Checked 5 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 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0
```

Ici la partie qui nous intéresse sont les deux dernière ligne :

- **Total Warnings: 0**
- **Total Erros: 0**

Ses deux lignes indique qu'ils y a aucune erreur.

si vous constater que les deux ligne sont différents que cela revoyez minutieusement votre configuration.

Voyons maintenant ce qui est à modifier dans le fichier `/usr/local/nagios/etc/objetc/localhost.cfg` et ainsi intégrer le serveur web au groupe de machine comme alias Linux Servers.

D'abord saisissez la commande **"nano /usr/local/nagios/etc/objetc/localhost.cfg"** . Nous allons simplement ajouter `srvweb` au membres.

```
define hostgroup {
    hostgroup_name    linux-servers      ; The name of the hostgroup
    alias             Linux Servers      ; Long name of the group
    members           localhost,srvweb   ; Comma separated list of hosts that belong to this group
}
```

sur la meme ligne memebbers, ajouter `srvweb` ce qui donne (`members localhost, srvweb`)

redémarrez le service nagios via la commande : `systemctl restart nagios`

The screenshot shows the Nagios web interface at `192.168.4.5/nagios/`. The interface includes a sidebar with navigation links and a main content area displaying network status and service overview.

Current Network Status
Last Updated: Fri Mar 27 21:00:46 CET 2026
Updated every 90 seconds
Nagios® Core™ 4.5.11 - www.nagios.org
Logged in as nagiosadmin

Host Status Totals

Up	Down	Unreachable	Pending
2	0	0	0

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
14	0	0	1	0

Service Overview For Host Group 'linux-servers'

Host	Status	Services	Actions
localhost	UP	7 OK 1 CRITICAL	[Icons]
srvweb	UP	7 OK	[Icons]

Ici ont constate que le `srvweb` fait bien partie du group: **linux-servers**

Etape 5 : Configuration des objets et services à superviser sur le serveur "SRVWEB" par le serveur "SRVNAGIOS"

Passons à la dernière étape la plus importante et la plus primordiale, la supervision des objets et services sur le serveur web.

Pour commencer aller sur le serveur web et configuration le fichier de `srvweb.cfg` en saisissant la commande **"nano /usr/local/nagios/etc/servers/srvweb.cfg"**. rempli ses le fichiers comme la capture le montre si de sous.

```
#Définition de la machine Linux "SRVWEB"

define host {
    use                linux-server
    host_name          srvweb
    alias              Service web Apache2
    address            192.168.4.4
}

#Déclaration d'un service/objet à superviser : Nbre utilisateurs connectés

define service {
    use                generic-service
    host_name          srvweb
    service_description Nbre utilisateurs
    check_command      check_nrpe!check_users!2!3
}
```

maintenant relancer le service nagios en saisissant la commande : `systemctl restart nagios`.

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
srvweb	Nbre utilisateurs	OK	03-11-2026 13:29:36	0d 0h 17m 22s	1/3	USERS OK - 1 users currently logged in
	Utilisation du disque	OK	03-11-2026 13:34:48	0d 0h 2m 10s	1/3	DISK OK - free space: / 15771 MiB (87.46% inode=96%): /dev 964 MiB (100.00% inode=100%): /dev/shm 983 MiB (100.00% inode=100%): /run 196 MiB (99.73% inode=100%): /run/lock 5 MiB (100.00% inode=100%): /run/user/1000 196 MiB (100.00% inode=100%): /run/credentials 196 MiB (99.73% inode=100%): /run/nrpe 196 MiB (99.73% inode=100%): /run/systemd/incoming 196 MiB (99.73% inode=100%): /tmp 15771 MiB (87.46% inode=96%): /var/tmp 15771 MiB (87.46% inode=96%):

Results 1 - 2 of 2 Matching Services

*mon disque est ok car j'ai fait déjà fait modifications mais normalement vous êtes sensé voire votre disque en unknow pour cela il faudra modifier le nom du disque que le srvweb doit donner, ainsi pour connaitre le nom de votre disque saisissez la commande **"sudo fdisk -l"**.

```
root@srvweb:/usr/local/nagios/libexec# sudo fdisk -l
Disque /dev/sda : 20 GiB, 21474836480 octets, 41943040 secteurs
Modèle de disque : VBOX HARDDISK
Unités : secteur de 1 × 512 = 512 octets
Taille de secteur (logique / physique) : 512 octets / 512 octets
taille d'E/S (minimale / optimale) : 512 octets / 512 octets
Type d'étiquette de disque : dos
Identifiant de disque : 0x99da4031
```

Pour ma part mon c'est disque c'est `"/dev/sda"`

maintenant éditons le fichier `nrpe.cfg` et corrigez la commande comme suis sur la capture

```
command[check_hda1]=/usr/local/nagios/libexec/check_disk -w 20% -c 10% -p /dev/sda
```

voici la commande correcte à saisir : **check_command check_nrpe!check_hda1!80!90**

```
#Déclaration d'un service/objet à superviser : capacité disque dur

define service {
    use                generic-service
    host_name          srvweb
    service_description Utilisation du disque
    check_command       check_nrpe!check_hda1!80!90
}
```

Dans votre fichier `nrpe.cfg` qui est sur le serveur la commande à modifier pour la prise en compte du nombre d'utilisateur est:

```
command[check_users]=/usr/local/nagios/libexec/check_users -w 2 -c 3
```

Explication :

1. `-w 2` le seuil d'utilisateur qui est 2 donc sa passe en warning
2. `-c 3` le nombre d'utilisateur critique plus de 3 = critical

Pour le disque ont veut que lorsque le disque à atteint 80% sa passe en warning et 90% en critical donc pareil mais sur la ligne de commande (`command[check_hda1]...`):

```
command[check_hda1]=/usr/local/nagios/libexec/check_users -w 80 -c 90 -p /dev/sda
```

Maintenant que vous avez compris ont va faire pareil pour les autres objets:

```

command[check_users]=/usr/local/nagios/libexec/check_users -w 2 -c 3
command[check_load]=/usr/local/nagios/libexec/check_load -r w .15,.10,.05 -c .30,.25,.20
command[check_hda1]=/usr/local/nagios/libexec/check_disk -w 20% -c 10% -p /dev/sda
command[check_zombie_procs]=/usr/local/nagios/libexec/check_procs -w 5 -c 10 -s 2
command[check_total_procs]=/usr/local/nagios/libexec/check_procs -w 150 -c 200

# 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.

### MISC SYSTEM METRICS ###
#command[check_users]=/usr/local/nagios/libexec/check_users $ARG1$
command[check_load]=/usr/local/nagios/libexec/check_load -w $ARG1$ -c $ARG2$
#command[check_disk]=/usr/local/nagios/libexec/check_disk $ARG1$
command[check_swap]=/usr/local/nagios/libexec/check_swap $ARG1$
command[check_cpu_stats]=/usr/local/nagios/libexec/check_cpu_stats.sh $ARG1$
command[check_mem]=/usr/local/nagios/libexec/check_swap -w $ARG1$ -c $ARG2$
command[check_apache]=/usr/local/nagios/libexec/check_procs -c 1: -C apache2
command[check_uptime]=/usr/local/nagios/libexec/check_uptime
command[check_http]=/usr/local/nagios/libexec/check_http -H localhost -p 80

```

Résulat:

Limit Results:

100

Host ↕	Service ↕	Status ↕	Last Check ↕	Duration ↕	Attempt ↕	Status Information
srvweb	Charge CPU	OK	03-27-2026 08:05:59	1d 22h 16m 32s	1/3	OK - load average: 0.00, 0.00, 0.00
	Charge Mémoire	OK	03-27-2026 08:05:59	1d 21h 3m 33s	1/3	SWAP OK - 100% free (974 MB out of 974 MB)
	Nbre utilisateurs	OK	03-27-2026 08:05:59	2d 4h 57m 29s	1/3	USERS OK - 1 users currently logged in
	Service Apache	OK	03-27-2026 08:05:59	1d 21h 16m 45s	1/3	PROCS OK: 6 processes with command name 'apache2'
	Service http	OK	03-27-2026 08:05:59	1d 20h 56m 7s	1/3	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.005 second response time
	Temps ecoule	OK	03-27-2026 08:05:59	1d 22h 27m 21s	1/3	Uptime OK: 1 day(s) 0 hour(s) 8 minute(s)
	Utilisation du disque	OK	03-27-2026 08:05:59	1d 20h 57m 21s	1/3	DISK OK - free space: /dev 964 MiB (100.00% inode=100%):

si service objet sont en unknow ou critical verifier chaque ligne de commande dans le fichier nrpe.cfg sur le serveur web et srvweb.cfg sur le serveur nagios

FIN