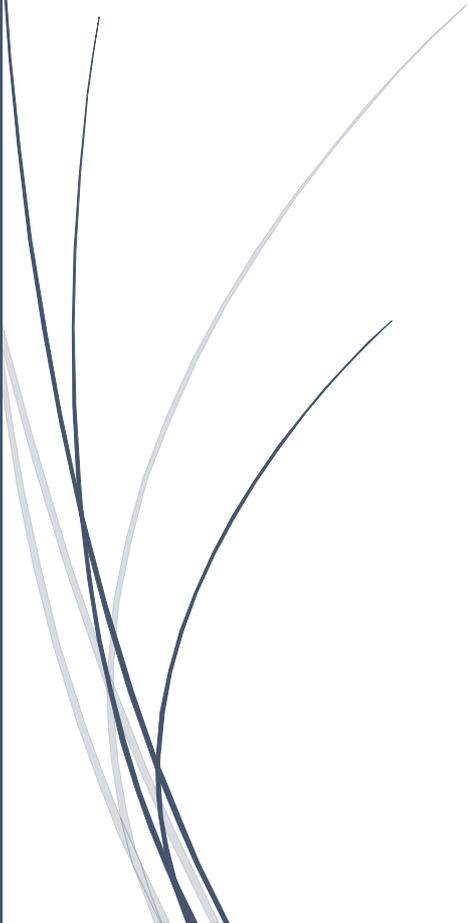


A dark blue vertical bar on the left side of the page. A blue arrow-shaped graphic points to the right from the bar, containing the text '2017-2018'.

2017-2018

Supervision de matériel informatique avec Shinken

Epreuve E6

Several thin, curved lines in shades of blue and grey originate from the bottom left and curve upwards and to the right.

Raphaël Andrieu
ARCONIC

Table des matières

Shinken	1
Prérequis	1
Installation de Shinken	1
Supervision	4
Installation de Nagios	4
Supervision d'un serveur Debian	4
Ajout d'hôte.....	5
Ajout du groupe de Serveur Linux.....	6
Ajout des services.....	6
Supervision d'un serveur Windows.....	7
Ajout de l'hôte	9
Ajout du groupe Serveur Windows	9
Ajout des services.....	10
Supervision d'une imprimante	12
Ajout d'un hôte.....	12
Ajout d'un groupe.....	12
Ajout d'un service.....	12
Supervision d'un Switch Cisco	13
Configuration sur le switch Cisco	14
Supervision d'un router.....	16
Ajout de l'hôte.....	16
Ajout du service.....	17

Shinken

Shinken est un logiciel libre permettant la surveillance système et réseau. Elle surveille les hôtes et services spécifiés (Serveur Linux, Windows, imprimante, router et switch Cisco), alertant lorsque les systèmes vont mal et quand ils vont mieux.

Prérequis

- Un switch Cisco
- Un serveur Windows 2012
- Deux serveurs Debian 8
- Une imprimante

Installation de Shinken

Avant d'installer Shinken, nous allons tout d'abord installer Python qui nous servira à faire fonctionner correctement Shinken qui est développé en python.

```
#apt-get update
#apt-get install python-pip python-pycurl
```

Puis nous allons ajouter un utilisateur Shinken pour pouvoir lancer l'installation de Shinken

```
#adduser shinken
```

Maintenant nous allons installer le service avec l'installateur de modules de Python (PIP)

```
#pip install shinken
```

Puis nous nous connectons en tant qu'utilisateur Shinken pour pouvoir installer et activer le module WEBUI2 qui nous permettra d'avoir une interface web pour le monitoring.

```
#su shinken
Shinken -init
Shinken install webui2
```

Ensuite, nous installons le système de gestion de base de données mongodb et le programme pip

```
Su root
#pip install pymongo>=3.0.3 requests arrow bottle==0.12.8
#apt-get install mongodb
```

Un fois le module PIP installé nous indiquons le module Webui2 dans le fichier broker-master.cfg

```
Modules webui2
```

```
shinken@srv-ar-deb: /home/andri
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
GNU nano 2.2.6  Fichier : /etc/shinken/brokers/broker-master.cfg  Modifié

# Default: None
# Interesting modules that can be used:
# - simple-log          = just all logs into one file
# - livestatus          = livestatus listener
# - tondodb-mysql      = NDO DB support (deprecated)
# - npcdmod             = Use the PNP addon
# - graphite           = Use a Graphite time series DB for perfdata
# - webui              = Shinken Web interface
# - glpidb             = Save data in GLPI MySQL database
modules      webui2

# Enable https or not
use_ssl      0
# enable certificate/hostname check, will avoid man in the middle attacks
hard_ssl_name_check  0

## Advanced
realm      All
}

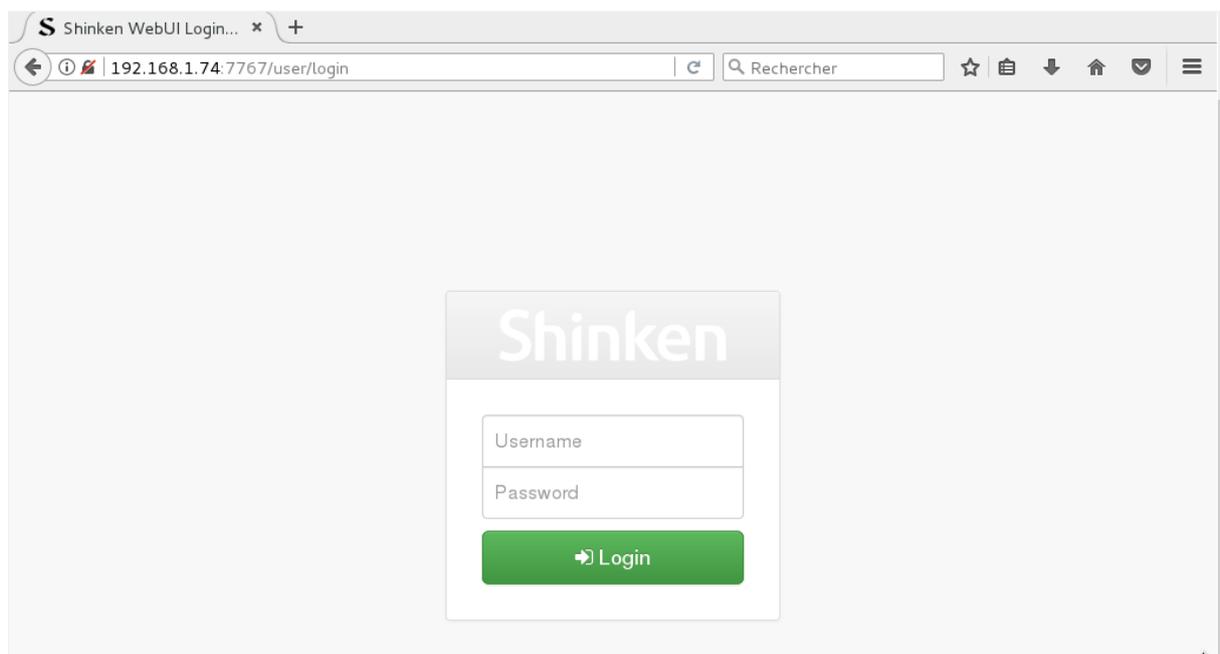
^G Aide      ^O Écrire    ^R Lire fich.^Y Page préc.^K Couper    ^C Pos. cur.
^X Quitter   ^J Justifier ^W Chercher  ^V Page suiv.^U Coller   ^T Orthograp.
```

Puis nous redémarrons le service Shinken

#Service shinken restart

Vous pouvez maintenant accéder a l'interface Web de Shinken avec l'adresse :

- <http://192.168.1.74:7767>



Supervision

Maintenant que notre serveur de supervision est lancé, nous allons pouvoir installer des plugins permettant de faire remonter les services principaux d'un serveur Windows, d'un switch Cisco et d'un serveur Linux. Pour cela nous allons installer le plugins Nagios qui nous permettra d'exécuter des commandes de récupération à distance.

Installation de Nagios

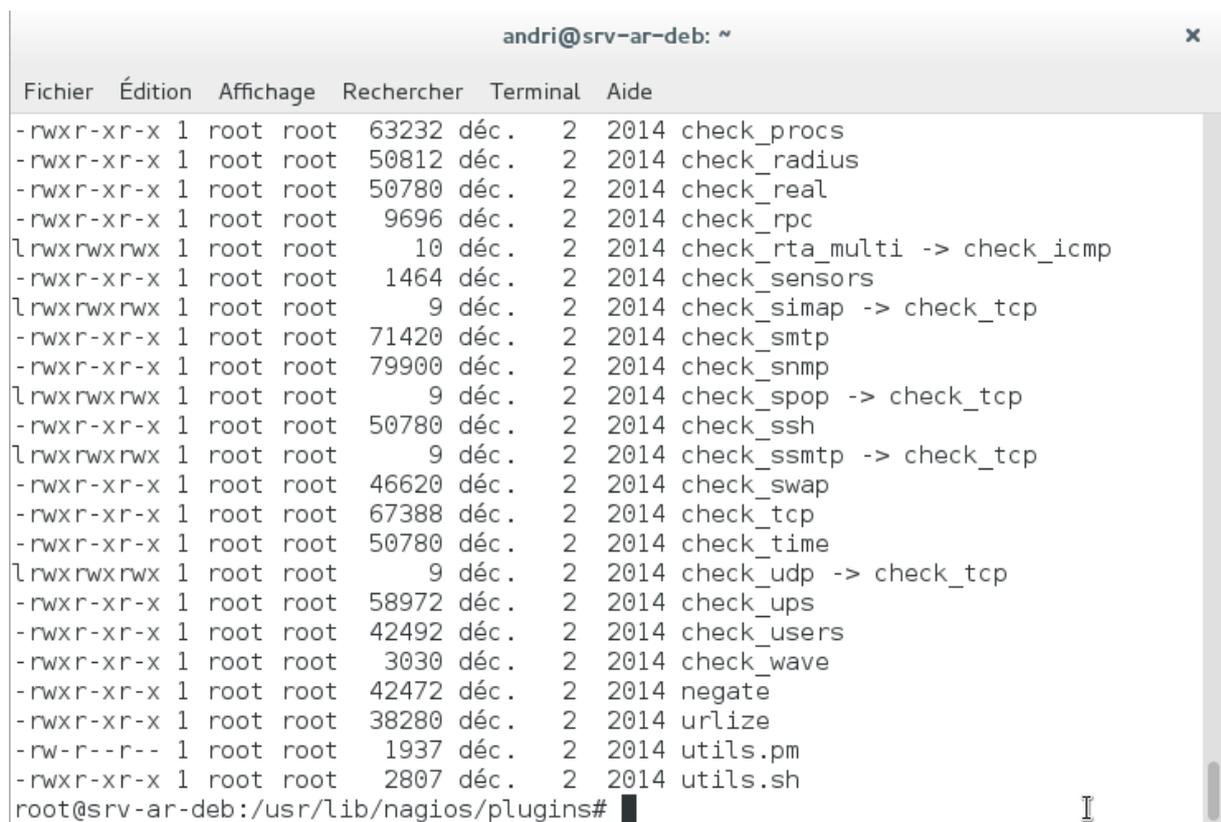
```
#apt-get install nagios-plugins
```

Maintenant que le plugin est installé, vous pouvez retrouver tous les plugins Nagios dans le dossier

« /usr/lib/nagios/plugins »

```
#cd /usr/lib/nagios/plugins
```

```
#ls -l
```



```
andri@srv-ar-deb: ~
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
-rwxr-xr-x 1 root root 63232 déc. 2 2014 check_procs
-rwxr-xr-x 1 root root 50812 déc. 2 2014 check_radius
-rwxr-xr-x 1 root root 50780 déc. 2 2014 check_real
-rwxr-xr-x 1 root root 9696 déc. 2 2014 check_rpc
lrwxrwxrwx 1 root root 10 déc. 2 2014 check_rta_multi -> check_icmp
-rwxr-xr-x 1 root root 1464 déc. 2 2014 check_sensors
lrwxrwxrwx 1 root root 9 déc. 2 2014 check_simap -> check_tcp
-rwxr-xr-x 1 root root 71420 déc. 2 2014 check_smtp
-rwxr-xr-x 1 root root 79900 déc. 2 2014 check_snmp
lrwxrwxrwx 1 root root 9 déc. 2 2014 check_spop -> check_tcp
-rwxr-xr-x 1 root root 50780 déc. 2 2014 check_ssh
lrwxrwxrwx 1 root root 9 déc. 2 2014 check_ssmtm -> check_tcp
-rwxr-xr-x 1 root root 46620 déc. 2 2014 check_swap
-rwxr-xr-x 1 root root 67388 déc. 2 2014 check_tcp
-rwxr-xr-x 1 root root 50780 déc. 2 2014 check_time
lrwxrwxrwx 1 root root 9 déc. 2 2014 check_udp -> check_tcp
-rwxr-xr-x 1 root root 58972 déc. 2 2014 check_ups
-rwxr-xr-x 1 root root 42492 déc. 2 2014 check_users
-rwxr-xr-x 1 root root 3030 déc. 2 2014 check_wave
-rwxr-xr-x 1 root root 42472 déc. 2 2014 negate
-rwxr-xr-x 1 root root 38280 déc. 2 2014 urlize
-rw-r--r-- 1 root root 1937 déc. 2 2014 utils.pm
-rwxr-xr-x 1 root root 2807 déc. 2 2014 utils.sh
root@srv-ar-deb:/usr/lib/nagios/plugins#
```

Supervision d'un serveur Debian

Maintenant que les plugins nagios sont installés, nous allons pouvoir superviser notre premier appareil, c'est-à-dire un serveur Debian ! Pour cela nous allons installer le plugins NRPE (Nagios Remote PluginExecutor) qui permet de superviser les serveurs Linux/Unix ou Windows a distance.

```
#apt-get install Nagios-nrpe-plugin
```

Puis sur notre serveur Debian

```
#apt-get install nagios-plugins
```

```
#apt-get install Nagios-nrpe-server
```

Nous pouvons maintenant modifier le fichier de configure de NRPE pour ajouter l'IP du serveur Shinken et autoriser les commandes de supervision.

```
#nano /etc/Nagios/nrpe.cfg
```

Puis modifier la ligne « `allowed_hosts=127.0.0.1` » en « `allowed_hosts=192.168.1.74` » (qui est l'IP de notre serveur Shinken, et décommentez la ligne « `command[check_disk]=/usr/lib/nagios/plugins/check_disk -w 10% -c 5% -p /` » qui nous permettra de récupérer les informations d'espace disque.

Maintenant nous allons rendre le démarrage du service NRPE automatique en utilisant le service « `chkconfig` » pour cela tapez les commandes

```
#apt-get install chkconfig
```

```
#chkconfig --add nagios-nrpe-server
```

Et nous allons redémarrer le service

```
# /etc/init.d/nagios-nrpe-server restart
```

Ajout d'hôte

Nous allons maintenant ajouter l'hôte du serveur linux sur le serveur Shinken, pour cela aller dans le répertoire « `/etc/shinken/hosts/` » et créez les fichiers `srv-deb-ar.cfg`

```
#cd /etc/shinken/hosts
```

```
Nano srv-deb-ar.cfg
```

```
define host{
    use         generic-host
    host_name   srv-ar-deb
    alias       srv-ar-deb
    address     192.168.1.200
}
```

```
#nano localhost.cfg
```

```
define host{
    use         generic-host
    contact_groups  admins
    address     127.0.0.1
    host_name   localhost
    address     localhost
}
```

Ajout du groupe de Serveur Linux

Nous allons maintenant créer un groupe pour ces serveurs Linux, ce qui nous permettra d'attribuer directement des services identiques pour éviter les redondances.

```
#cd /etc/shinken/hostgroups
```

```
#nano linux.cfg
```

Puis décommenter la ligne « members » et ajoutez les membres « localhost » et « srv-ar-deb »

```
define hostgroup{
    hostgroup_name linux ; The name of the hostgroup
    alias Linux Servers ; Long name of the group
    members localhost,srv-ar-deb
}
```

I

Ajout des services

Nous allons maintenant ajouter des services, pour cela aller dans le dossier « /etc/shinken/services »

```
#cd /etc/shinken/services
```

```
#nano linux.cfg
```

```
#Check DISK
define service{
    use generic-service
    hostgroup_name linux
    service_description Disk
    check_command check_nrpe!check_disk
}

#Check PING
define service{
    use generic-service
    hostgroup_name linux
    service_description PING
    check_command check_ping!100.0,20%!500.0,60%
}

#Check Users
define service{
    use generic-service
    hostgroup_name linux
    service_description Users
    check_command check_nrpe!check_users
}
```

Nous avons ajouté 3 commandes, une pour récupérer les informations d'espace disque, un pour savoir si le serveur est pingable et un autre pour récupérer l'utilisateur connecté.

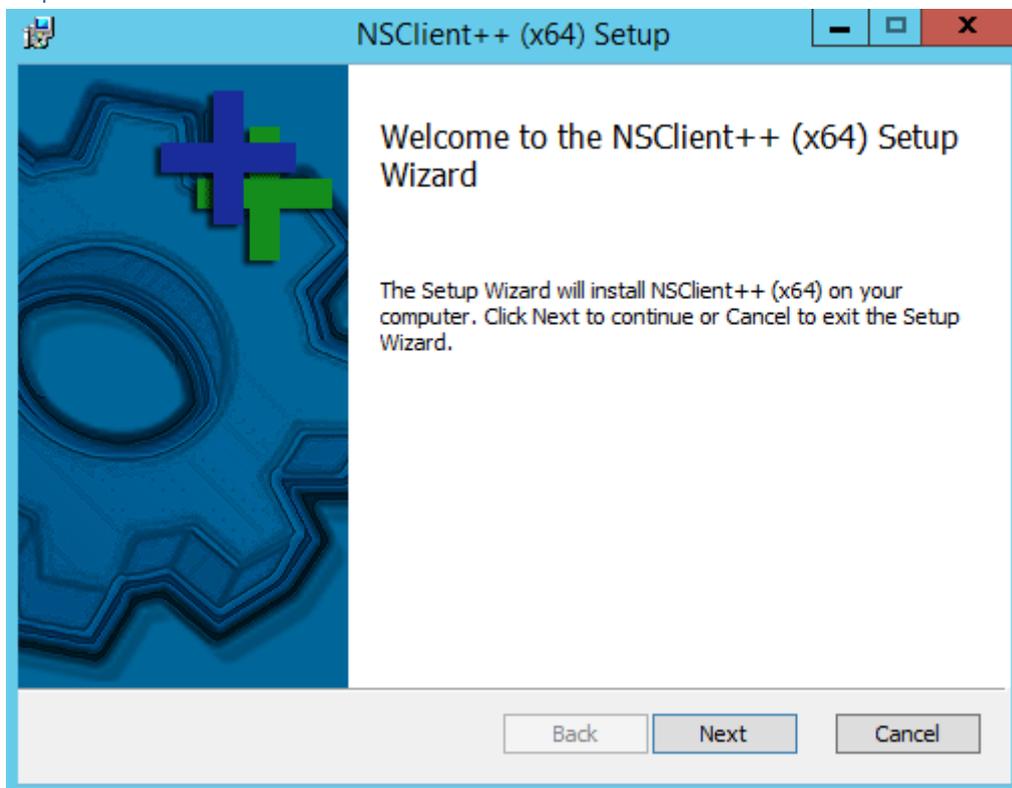
Nous redémarrons Shinken

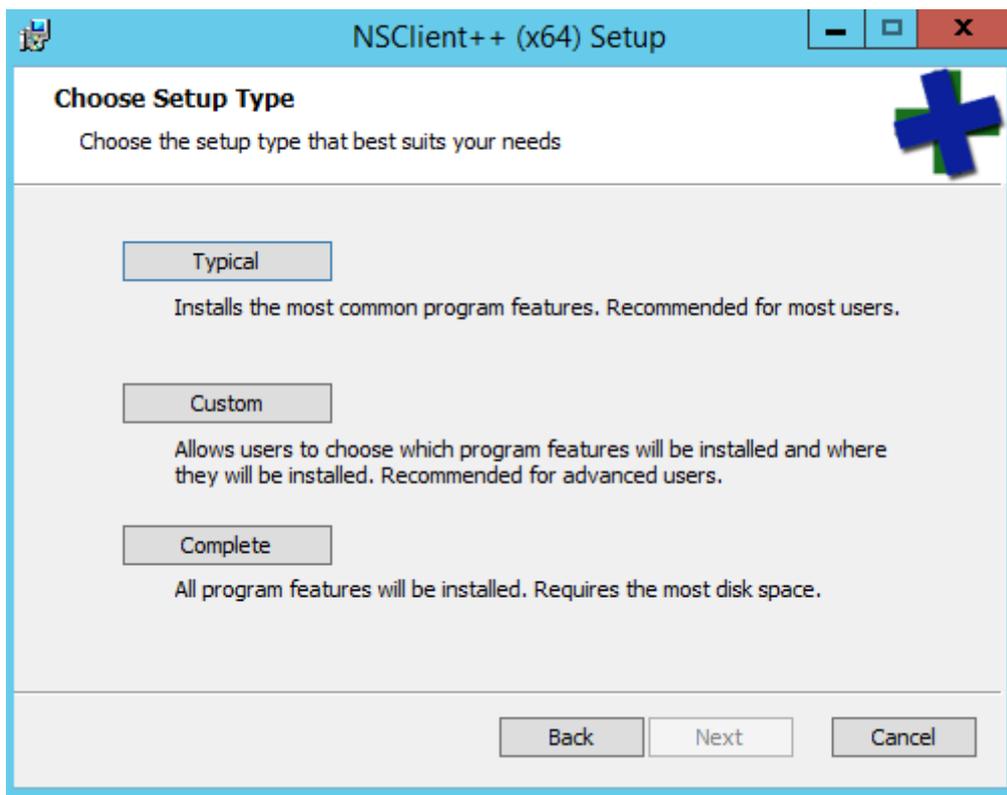
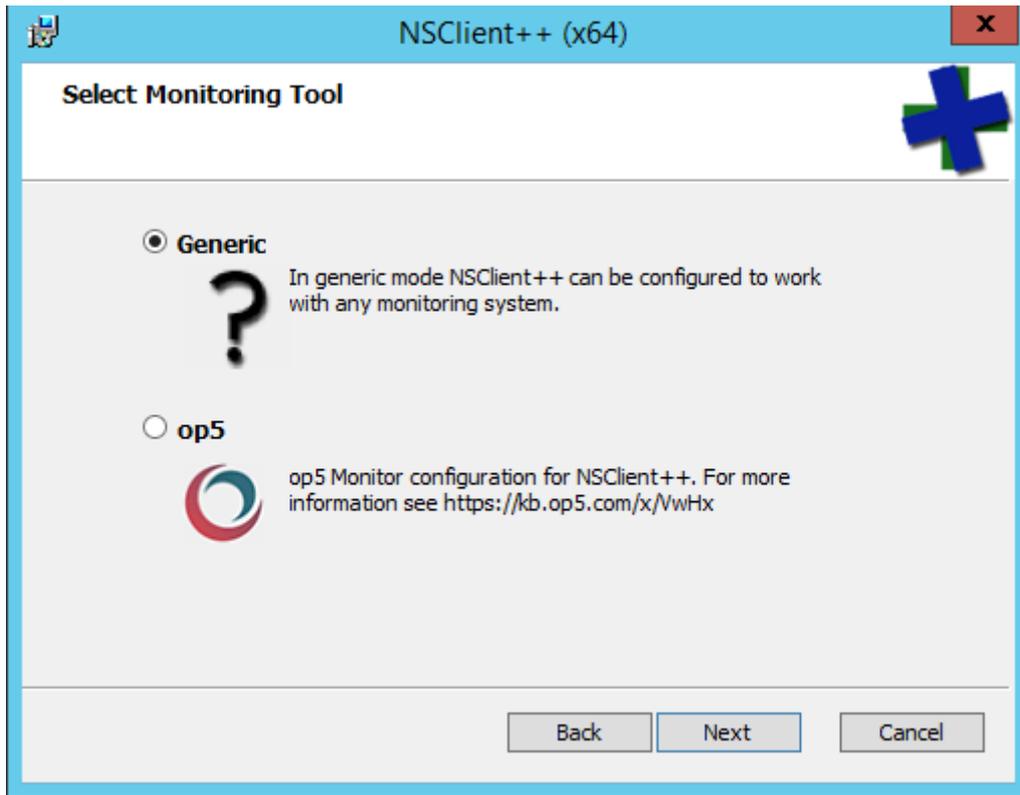
```
#service shinken restart
```

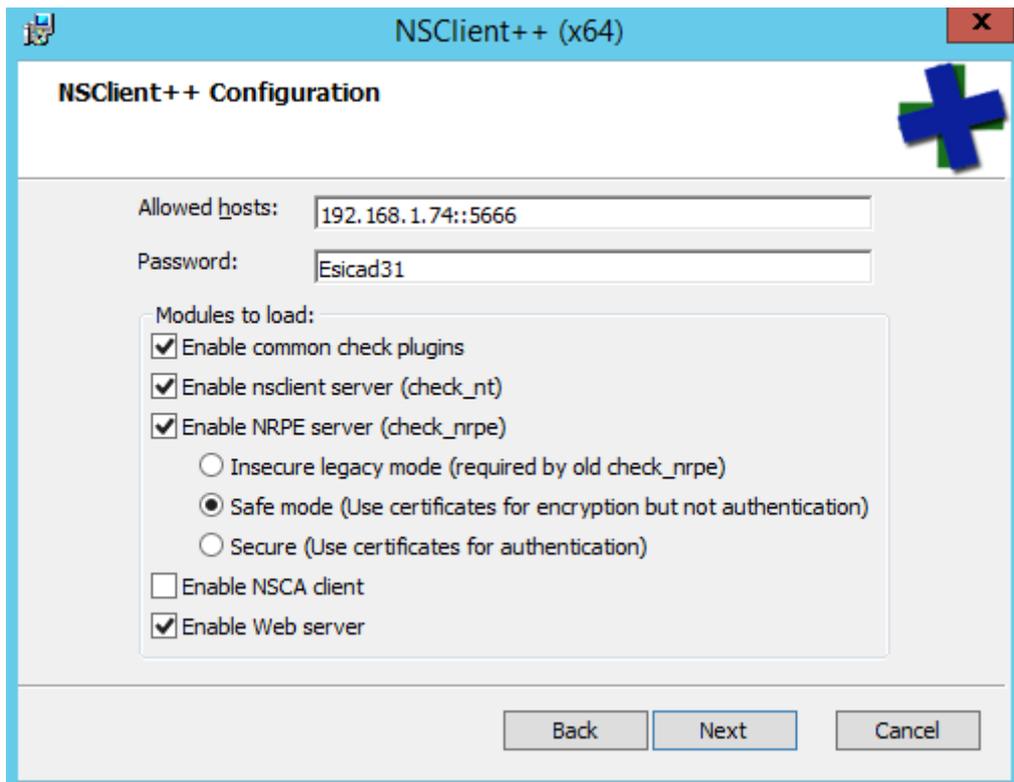
The screenshot displays the Nagios XI monitoring interface. At the top, there are summary statistics: 1 host (1/2 OK), 3 services (3/4 OK), and various other status indicators. Below this, the 'Business impact' is listed as 'Normal'. The main section shows a list of services for 'srv-ar-deb' on host 'srv-ar-deb':

Service Name	Status	Check Command	Output
UP	19s	srv-ar-deb	PING OK - Packet loss = 0%, RTA = 0...
OK	21s	srv-ar-deb / Disk	DISK OK - free space: / 3348 MB (46%...)
OK	51s	srv-ar-deb / PING	OK - 192.168.1.200: rta 0.313ms, los...
OK	3m 49s	srv-ar-deb / Users	UTILISATEURS OK - 2 utilisateurs actuel...

Supervision d'un serveur Windows







On ajoute l'ip de notre serveur Shinken avec le port 5666 qui est le port NRPE, puis nous ajoutons un mot de passe « Esicad31 »

Ajout de l'hôte

```
#cd /etc/shinken/hosts
```

```
#nano srv-lipton.cfg
```

```
define host{
    use      generic-host
    host_name      srv-lipton
    alias      srv-lipton
    address 192.168.1.205
}
```

Ajout du groupe Serveur Windows

```
#cd /etc/shinken/hostgroups
```

```
#nano windows.cfg
```

```
define hostgroup{
    hostgroup_name windows ;          The name of the hostgroups
    alias    Windows Servers ; Long name of the group
    members srv-lipton
}
```

Ajout des services

Nous allons vérifier que la connexion se fait bien entre notre serveur Windows et notre serveur Shinken, pour cela taper la commande

```
# /usr/lib/nagios/plugins/check_nt -H 192.168.1.205 -v CLIENTVERSION -p 12489 -s Esicad31
```

```
root@srv-ar-shinken:/usr/local/lib# /usr/lib/nagios/plugins/check_nt -H 192.168.1.205 -v CLIENTVERSION -p 12489 -s Esicad31
NSClient++ 0.5.2.35 2018-01-28
root@srv-ar-shinken:/usr/local/lib#
```

La réponse du serveur : « NSClient++ 0.5.2.35 », ce qui montre que tout est okay ! Si le serveur vous répond « Connexion refusée », il faut vérifier le pare-feu Windows et activer le port utilisé.

Nous allons maintenant créer une nouvelle commande « check_nt » dans Shinken en mettant le mot de passe du serveur Windows et le port utilisé.

```
#cd /etc/shinken/commands
```

```
#nano check_nt.cfg
```

```
define command{
    command_name    check_nt
    command_line    $NAGIOSPLUGINDIR$/check_nt -H $HOSTADDRESS$ -p 12489 -s Esicad31 -v $ARG1$ $ARG2$
}
```

Puis nous créons les services Windows à superviser

```
#cd /etc/shinken/services
```

```
#nano windows.cfg
```

```

#Check PING
define service{
    use generic-service
    host_name      srv-lipton
    service_description  PING
    check_command  check_ping!100.0,20%500.0,60%
}
█
#Check Client Version
define service{
    use generic-service
    host_name      srv-lipton
    service_description  Version NSClient++
    check_command  check_nt!CLIENTVERSION
}

#Check Disk
define service{
    use generic-service
    host_name      srv-lipton
    service_description  Espace Disque
    check_command  check_nt!USEDISKSPACE!-l c -w 80 -c 90
}

```

↑

Puis nous redémarrons le serveur

```
#/etc/init.d/shinken restart
```

```

Restarting scheduler
. ok
Restarting poller
. ok
Restarting reactionner
. ok
Restarting broker
. ok
Restarting receiver
. ok
Restarting arbiter
Doing config check
. ok
. ok

```

↑

Supervision d'une imprimante

Ajout d'un hôte

Nous allons maintenant superviser une imprimante pour savoir si elle est bien allumée et accessible sur le réseau. Pour cela nous allons créer l'hôte « ompr-canon.cfg »

```
#cd /etc/shinken/hosts
```

```
#nano impr-canon.cfg
```

```
define host{
    use         generic-host
    host_name   impr-canon
    alias       srv-impr-canon
    address     192.168.1.78
}
```

Ajout d'un groupe

Puis nous allons créer un groupe imprimante pour y ajouter notre imprimante canon

```
#cd /etc/shinken/hostgroups
```

```
#nano imprimante.cfg
```

```
define hostgroup{
    hostgroup_name  imprimante;
    alias           imprimante;
    members         impr-canon
}
```

Ajout d'un service

Puis nous créons un service pour savoir si l'imprimante est pingable ou non.

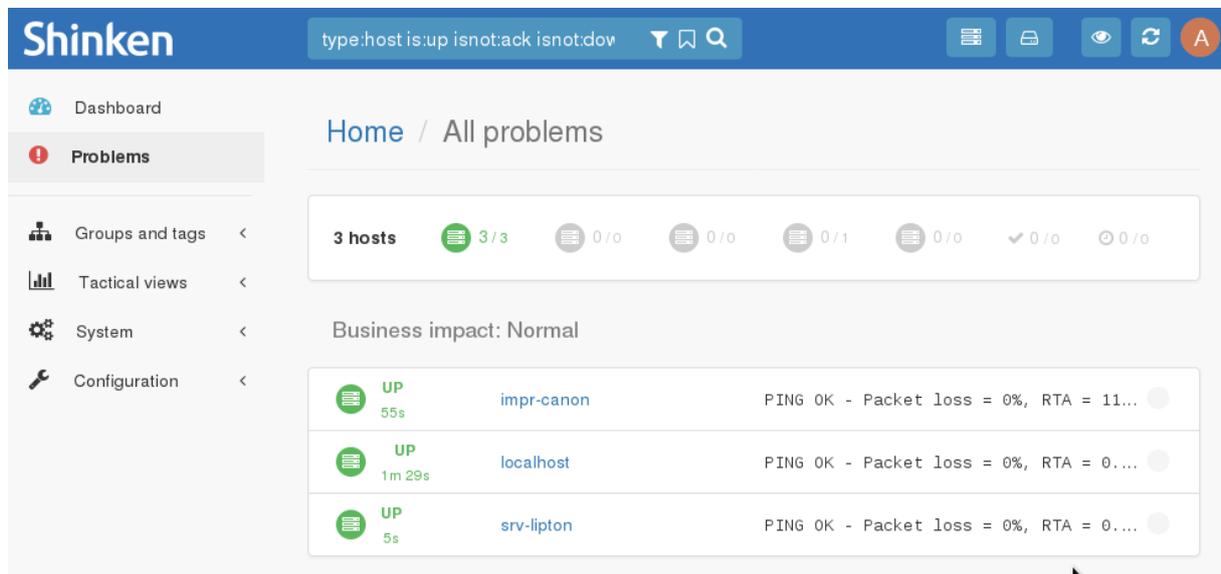
```
#cd /etc/shinken/services
```

```
#nano imprimante.cfg
```

```
#Check PING
define service{
    use          generic-service
    host_name    impr-canon
    service_description    PING
    check_command    check_ping!100.0,20%!500.0,60%
}
}
```

Puis nous redémarrons le service

```
#!/etc/init.d/shinken restart
```



The screenshot shows the Shinken web interface. The top navigation bar includes the Shinken logo, a search bar with the text 'type:host is:up isnot:ack isnot:dov', and several utility icons. The left sidebar contains a menu with 'Dashboard', 'Problems', 'Groups and tags', 'Tactical views', 'System', and 'Configuration'. The main content area displays 'Home / All problems' and a summary for '3 hosts' with a status of '3/3' (all green). Below this, the 'Business impact' is shown as 'Normal'. A table lists three hosts:

Host	Status	Details
impr-canon	UP (55s)	PING OK - Packet loss = 0%, RTA = 11...
localhost	UP (1m 29s)	PING OK - Packet loss = 0%, RTA = 0...
srv-lipton	UP (5s)	PING OK - Packet loss = 0%, RTA = 0...

Supervision d'un Switch Cisco

Nous allons maintenant nous occuper de superviser un Switch Cisco 3750 séries, pour cela nous allons installer les packs « cisco » et « switch » sur notre serveur Shinken

```
#Su shinken
```

```
shinken install cisco
```

```
shinken install switch
```

```
^X^Cshinken@srv-ar-shinken:/etc/shinken/services$ shinken install cisco
Grabbing : cisco
OK cisco
shinken@srv-ar-shinken:/etc/shinken/services$ shinken install switch
Grabbing : switch
OK switch
shinken@srv-ar-shinken:/etc/shinken/services$ █
```

Nous allons maintenant pouvoir configurer le plugin, pour cela nous allons installer la commande « check_nwc_health » pour récupérer des infos sur le switch cisco, comme sa mémoire, son hardware, son processeur, et sur ses interfaces.

```
su root
#cd /var/lib/shinken
#wget https://labs.consol.de/assets/downloads/nagios/check_nwc_health-7.0.1.1.tar.gz
#tar -xzf check_nwc_health-7.0.1.1.tar.gz
#cd check_nwc_health-7.0.1.1
```

Nous allons maintenant pouvoir compiler le plugin, pour cela vous pouvez vous référer au README en faisant « nano readme »

```
#!/configure --prefix=/var/lib/shinken/ --with-nagios-user=shinken --with-nagios-group=shinken --withperl=/usr/bin/perl
#make
#make install
```

Configuration sur le switch Cisco

Nous allons configurer le SNMP (Simple Network Management Protocol) sur le switch, pour cela connectez vous à l'aide de putty sur le switch, et tapez les commandes suivantes :

```
Switch>en
Switch# conf t
Switch(config)#snmp-server community public RO
```

Nous allons maintenant pouvoir tester si la communication se fait bien entre notre serveur Shinken et le Switch Cisco, pour cela tapez la commande

```
#!/var/lib/shinken/libexec/check_nwc_health --hostname 192.168.1.209 --timeout 60 --community "public" --mode interface-status
```

Cette commande devrait retourner le statut de toutes les interfaces de notre switch

```

ition is presumed to exist on FastEthernet2/0/27 (alias === PCS ===), fault condition is presumed to exist on Fa
stEthernet2/0/28 (alias === PCS ===), fault condition is presumed to exist on FastEthernet2/0/29 (alias === PCS
===), fault condition is presumed to exist on FastEthernet2/0/30 (alias === PCS ===), fault condition is presume
d to exist on FastEthernet2/0/32, fault condition is presumed to exist on FastEthernet2/0/33, fault condition is
presumed to exist on FastEthernet2/0/34, fault condition is presumed to exist on FastEthernet2/0/35, fault cond
ition is presumed to exist on FastEthernet2/0/36, fault condition is presumed to exist on FastEthernet2/0/37, fau
ult condition is presumed to exist on FastEthernet2/0/38, fault condition is presumed to exist on FastEthernet2/
0/39, fault condition is presumed to exist on FastEthernet2/0/40, fault condition is presumed to exist on FastEt
hernet2/0/41, fault condition is presumed to exist on FastEthernet2/0/42, fault condition is presumed to exist o
n FastEthernet2/0/43, fault condition is presumed to exist on FastEthernet2/0/44, fault condition is presumed to
exist on FastEthernet2/0/45, fault condition is presumed to exist on FastEthernet2/0/46, fault condition is pre
sumed to exist on FastEthernet2/0/47, fault condition is presumed to exist on FastEthernet2/0/48 (alias R1), fau
lt condition is presumed to exist on GigabitEthernet2/0/1, fault condition is presumed to exist on GigabitEthern
et2/0/2, fault condition is presumed to exist on GigabitEthernet2/0/3, fault condition is presumed to exist on G
igabitEthernet2/0/4, Vlan1 is down/down, StackPort2 is down/up, StackSub-St2-1 is down/up, StackSub-St2-2 is dow
n/up, FastEthernet2/0/1 (alias === BUSINESS ===) is down/up, FastEthernet2/0/2 (alias === BUSINESS ===) is down/
up, FastEthernet2/0/3 (alias === BUSINESS ===) is down/up, FastEthernet2/0/4 (alias === BUSINESS ===) is down/up
, FastEthernet2/0/5 (alias === BUSINESS ===) is down/up, FastEthernet2/0/6 (alias === BUSINESS ===) is down/up,
FastEthernet2/0/7 (alias === BUSINESS ===) is down/up, FastEthernet2/0/8 (alias === BUSINESS ===) is down/up, Fa
stEthernet2/0/9 (alias === BUSINESS ===) is down/up, FastEthernet2/0/10 (alias === BUSINESS ===) is down/up, Fas
tEthernet2/0/11 (alias === BUSINESS ===) is down/up, FastEthernet2/0/12 (alias === BUSINESS ===) is down/up, Fas
tEthernet2/0/13 (alias === BUSINESS ===) is down/up, FastEthernet2/0/14 (alias === BUSINESS ===) is down/up, Fas
tEthernet2/0/15 (alias === BUSINESS ===) is down/up, FastEthernet2/0/16 (alias === PCS ===) is down/up, FastEthe
rnet2/0/17 (alias === PCS ===) is down/up, FastEthernet2/0/18 (alias === PCS ===) is down/up, FastEthernet2/0/19
(alias === PCS ===) is down/up, FastEthernet2/0/20 (alias === PCS ===) is down/up, FastEthernet2/0/21 (alias ==
= PCS ===) is down/up, FastEthernet2/0/22 (alias === PCS ===) is down/up, FastEthernet2/0/23 (alias === PCS ===)
is down/up, FastEthernet2/0/24 (alias === PCS ===) is down/up, FastEthernet2/0/25 (alias === PCS ===) is down/u
p, FastEthernet2/0/26 (alias === PCS ===) is down/up, FastEthernet2/0/27 (alias === PCS ===) is down/up, FastEth
ernet2/0/28 (alias === PCS ===) is down/up, FastEthernet2/0/29 (alias === PCS ===) is down/up, FastEthernet2/0/3
0 (alias === PCS ===) is down/up, FastEthernet2/0/31 is up/up, FastEthernet2/0/32 is down/up, FastEthernet2/0/33
is down/up, FastEthernet2/0/34 is down/up, FastEthernet2/0/35 is down/up, FastEthernet2/0/36 is down/up, FastEt
hernet2/0/37 is down/up, FastEthernet2/0/38 is down/up, FastEthernet2/0/39 is down/up, FastEthernet2/0/40 is dow
n/up, FastEthernet2/0/41 is down/up, FastEthernet2/0/42 is down/up, FastEthernet2/0/43 is down/up, FastEthernet
/0/44 is down/up, FastEthernet2/0/45 is down/up, FastEthernet2/0/46 is down/up, FastEthernet2/0/47 is down/up, F
astEthernet2/0/48 (alias R1) is down/up, GigabitEthernet2/0/1 is down/up, GigabitEthernet2/0/2 is down/up, Gigab
itEthernet2/0/3 is down/up, GigabitEthernet2/0/4 is down/up, Null0 is up/up
root@srv-ar-shinken:/var/lib/shinken/libexec#

```

Maintenant que nous avons vérifié que le serveur communique bien avec le switch, nous allons pouvoir ajouter un nouvel hôte

```
#cd /etc/shinken/hosts/
```

```
#nano switch.cfg
```

```

define host{
    use        switch
    contact_groups admins
    host_name Cisco Catalyst 3750 Series
    address 192.168.1.209
}

```

Puis nous redémarrons notre serveur Shinken

```
#!/etc/init.d/shinken restart
```

2 hosts 2/2 0/0 0/3 0/0 0/0 0/0 0/0 0/0

Business impact: Normal

UP 2m 28s	Cisco Catalyst 3750 Series	PING OK - Packet loss = 0%, RTA = 1...
UP 1m 26s	localhost	PING OK - Packet loss = 0%, RTA = 0...

host: Cisco Catalyst 3750 Series

```
(alias === PCS ===) is down/up,
FastEthernet2/0/26 (alias === PCS ===) is
down/up, FastEthernet2/0/27 (alias ===
PCS ===) is down/up, FastEthernet2/0/28
(alias === PCS ===) is down/up,
FastEthernet2/0/29 (alias === PCS ===) is
down/up, FastEthernet2/0/30 (alias ===
PCS ===) is down/up, FastEthernet2/0/31
is up/up, FastEthernet2/0/32 is down/up,
FastEthernet2/0/33 is down/up,
FastEthernet2/0/34 is down/up,
FastEthernet2/0/35 is down/up,
FastEthernet2/0/36 is down/up,
FastEthernet2/0/37 is down/up,
FastEthernet2/0/38 is down/up,
FastEthernet2/0/39 is down/up,
FastEthernet2/0/40 is down/up,
FastEthernet2/0/41 is down/up,
FastEthernet2/0/42 is down/up,
FastEthernet2/0/43 is down/up,
FastEthernet2/0/44 is down/up,
FastEthernet2/0/45 is down/up,
FastEthernet2/0/46 is down/up,
FastEthernet2/0/47 is down/up,
FastEthernet2/0/48 (alias R1) is down/up,
GigabitEthernet2/0/1 is down/up
```

Supervision d'un router

Ajout de l'hôte

```
# cd /etc/shinken/hosts
```

```
# nano sfr-box.cfg
```

```
define host{
    use      generic-host
    host_name    Sfr-box
    alis      Box SFR
    address 192.168.1.1
}
```

Ajout du service

```
#/etc/shinken/services/sfr-box.cfg
```

```
#Check PING
define service{
    use      generic-service
    host_name    Sfr-box
    service_description    PING
    check_command    check_ping!100.0,20%!500.0,60%
}
```

Puis nous redémarrons le service

```
#/etc/init.d/shinken restart
```

The screenshot shows the Shinken monitoring dashboard for host: Sfr-box. The interface includes a navigation menu on the left with options like Dashboard, Problems, Groups and tags, Tactical views, System, and Configuration. The main content area displays the following information:

- Home / All problems**
- 1 hosts** status: 1/3 UP, 0/0 DOWN, 0/3 CRITICAL, 0/0 UNREACHABLE, 0/0 UNKNOWN, 0/0 OK.
- 1 services** status: 1/2 UP, 0/0 DOWN, 0/3 CRITICAL, 0/2 UNREACHABLE, 0/7 UNKNOWN, 0/0 OK.
- Business impact: Normal**
- UP 3s Sfr-box**: PING OK - Packet loss = 0%, RTA = 0...
- OK 3s Sfr-box / PING**: OK - 192.168.1.1: rta 1.754ms, lost 0%