

## Table des matières

1. Relancer le processus Apache sous Linux .....	1
2.....	5

### 1. Relancer le processus Apache sous Linux

Pour pouvoir relancer le processus Apache automatiquement s'il tombe nous devons créer une commande qui va relancer le service manuellement si notre serveur centreon remarque un état critique sur notre service de check apache

Pour se faire nous devons nous connecter au serveur centreon avec l'utilisateur centreon-engine :

```
su - centreon-engine
```

Puis générer une paire de clé publique/privée ssh :

```
ssh-keygen
```

Et copier la clé publique sur notre machine debian qui tient le serveur Apache :

```
ssh-copy-id -i $USER$@<IP Serveur Debian>
```

**/ ! \ Attention à bien remplacer le \$USER\$ par le nom d'utilisateur de la machine debian et l'IP par son IP**

Maintenant nous devons aller sur notre machine debian et éditer le fichier /etc/sudoers.d/\$USER\$ :

**/ ! \ Attention à bien remplacer le \$USER\$ par le nom d'utilisateur de la machine debian**

```
# Debian Engine
```

```
$USER$ ALL = NOPASSWD: /usr/sbin/service apache2 restart
```

Editer ce fichier va permettre de ne pas donner le mot de passe lors du redémarrage du service apache par l'utilisateur

Nous pouvons maintenant tester une commande qui vérifie l'état du service Apache :

```
#APACHE ETEINT
```

```
[centreon-engine@localhost plugins]$ /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 /usr/sbin/apache2' -c 5:1024' -E
```

```
PROCS CRITICAL: 0 processes with args '/usr/sbin/apache2' | procs=0;;5:1024;0;
```

```
#APACHE ALLUME
```

```
[centreon-engine@localhost plugins]$ /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 /usr/sbin/apache2' -c 5:1024' -E
```

```
PROCS OK: 6 processes with args '/usr/sbin/apache2' | procs=6;;5:1024;0;
```

-c : Défini le seuil critique (ici moins de 5 processus apache on passe en état critique)

-i : Défini le chemin vers la clé privée du serveur centreon

-l : Défini le nom d'utilisateur sur la machine à laquelle on se connecte en SSH

-C : Défini la commande que l'on passe sur la machine

Une fois que l'on sait si notre service Apache fonctionne on peut lancer une commande qui redémarre le service s'il tombe :

```
[centreon-engine@localhost plugins]$ /usr/bin/ssh debian@192.168.7.1 "sudo /usr/sbin/service apache2 restart"
```

```
+-----+
LINUXVMIMAGES.COM
+-----+
User Name: debian
Password: debian (sudo su -)
```

On voit bien qu'il ne nous a pas demandé de mot de passe pour le redémarrage du service

### Passage des commandes dans centreon :

Maintenant que nous avons des commandes fonctionnelles nous pouvons les mettre en place dans centreon

Tout d'abord il faut créer les commandes :

Commande en Event Handler :

| Modify a Command

Check

Command Name \* handler\_ssh

Command Type ☐ Notification ☒ Check ☐ Misc ☐ Discovery

Command Line \* /usr/bin/ssh debian@\$HOSTADDRESS\$ "sudo /usr/sbin/service apache2 restart"

Commande de check by SSH :

| Modify a Command

Check

Command Name \* check\_by\_ssh




Command Type ☐ Notification ☒ Check ☐ Misc ☐ Discovery

Command Line \* \$LOCAL\$/check\_by\_ssh -H \$HOSTADDRESS\$ -i \$SSH\$ -l \$HOSTNAME\$ -C '\$LOCAL\$/check\_procs -a '/usr/sbin/apache2' -c 4:1024' -E



Ensuite il faut créer un template de service qui va être lié à notre commande de check by ssh :

| Modify a Service Template Model


General Information

Alias *	check_apache_ssh
Name *	check_apache_ssh
Template	Template   

Service Check Options




Check Command	check_by_ssh  				
Custom macros	<div> <div><input type="checkbox"/></div> Template inheritance         <div><input type="checkbox"/></div> Command inheritance       </div> <div> <a href="#">+ Add a new entry</a>          Nothing here, use the "Add" button       </div>				
Args	<table border="1"> <thead> <tr> <th>Argument</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td colspan="2">No argument found for this command</td> </tr> </tbody> </table>	Argument	Value	No argument found for this command	
Argument	Value				
No argument found for this command					

Service Scheduling Options

Check Period	24x7 
Max Check Attempts	3
Normal Check Interval	<input type="text"/> * 60 seconds
Retry Check Interval	<input type="text"/> * 60 seconds
Active Checks Enabled	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Default
Passive Checks Enabled	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Default
Is volatile	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Default

Dans l'onglet "Data Processing" nous devons activer l'événement handler et y associer notre commande créée précédemment :

Event Handler

Event Handler Enabled	<input checked="" type="radio"/> Yes <input type="radio"/> No <input type="radio"/> Default
Event Handler	handler_ssh  
Args	<input type="text"/> 

**/ ! \ Attention : Si à la fin de la manipulation le service ne remonte pas automatiquement il se peut qu'il faille faire la manipulation juste au-dessus dans l'onglet Data Processing du service by host**

Nous devons maintenant créer un template d'hôte et le lier à notre template de service :

| Modify a Host Template

Host basic information

Name *	template-debian
Alias *	template-debian

La liaison au template de service se fait dans l'onglet relations :

| Modify relations

Hostgroup Relations

Linked Service Templates	<div>  check_apache_ssh       </div>  
--------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Il ne nous reste plus qu'à lier notre template d'hôte à notre hôte debian :

[| Modify a Host](#)

Host basic information

Name *	debian
Alias	debian
IP Address / DNS *	192.168.7.1 <span>Resolve</span>
SNMP Community & Version	<input type="text"/> <span>▼</span>
Monitored from	Central <span>▼</span>
Timezone / Location	Timezone / Location <span>▼</span> <span>⊗</span>
Templates	<div>+ Add a new entry</div> <div>template-debian <span>▼</span> <span>⊕</span> <span>✎</span> <span>⊗</span></div>
Create Services linked to the Template too	<input type="radio"/> Yes <input checked="" type="radio"/> No

Nous devons exporter la configuration puis nous rendre dans l'onglet Monitoring pour procéder aux tests :

OK check\_apache\_ssh ⚙

● debian

⋮ 📄

DETAILS

TIMELINE

GRAPH

Command 📄

```
/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  
/usr/sbin/apache2' -c 4:1024'  
-E
```

Status information

PROCS OK: 6 processes with args '/usr/sbin/apache2'

Notre service apache est fonctionnel nous pouvons maintenant le "tuer" :

```
root@debian10:/usr/lib/centreon/plugins# kill apache2
```

On voit bien que le service n'est plus en "active running"

```
root@debian10:/usr/lib/centreon/plugins# systemctl status apache2  
  
● apache2.service - The Apache HTTP Server  
Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)  
Active: deactivating (stop) since Thu 2022-01-13 09:52:09 CST; 2s ago  
Docs: https://httpd.apache.org/docs/2.4/  
Process: 6133 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)  
Main PID: 6137 (code=exited, status=0/SUCCESS); Control PID: 6460 (apachectl)  
Tasks: 2 (limit: 4915)
```

*Memory: 3.7M*

*CGroup: /system.slice/apache2.service*

*└─6460 /bin/sh /usr/sbin/apachectl stop*

*└─6462 /usr/sbin/apache2 -k stop*

*Jan 13 08:54:57 debian10.linuxvmimages.local systemd[1]: Starting The Apache HTTP Server...*

*Jan 13 08:54:57 debian10.linuxvmimages.local systemd[1]: Started The Apache HTTP Server.*

On voit aussi le service tomber sur le serveur :

 **CRITICAL** S check\_apache\_ssh debian 5s 1/3 (S) 5s PROCS CRITICAL: 0 processes with args '/usr/sbin/apache2'

Et si nous relançons un autre check quelques secondes après nous voyons que le service est remonté automatiquement :

 **OK** S check\_apache\_ssh debian 8s 2/3 (S) 8s PROCS OK: 6 processes with args '/usr/sbin/apache2'

2.