

## Monitorer un service sur le serveur local

Le but est de savoir si un service est actif ou non. Comme si on entrait la commande  
« `systemctl status nomduserice.service` »

On va donc installer le script prévu à cet effet nommé `check_service.sh` sur [github](#)

Et ensuite l'installer sur la machine linux dans le répertoire « `/usr/local/nagios/libexec` » on peut aussi copier le script et le coller dans un fichier « `check_service.sh` » crée à l'occasion puis effectuer un `sudo chmod 755 ./check_service.sh` pour le rendre exécutable.

Maintenant que le script est installé il faudra en créer une commande nagios pour qu'il puisse la lire lors de ses checks. À chaque check nagios lis ses fichiers de configuration dont le fichier « `commands.cfg` » ou « `localhost.cfg` » où nous allons entrer la commande pour vérifier l'état d'un service.

On se rend donc dans le fichier `commands.cfg` dans le repertoire suivant :

```
tbagne@testbian:~$ cd /usr/local/nagios/etc/objects/  
tbagne@testbian:/usr/local/nagios/etc/objects$ ls
```

### sudo nano commads.cfg

on crée une commande comme ceci :

```
define command {  
    command_name      check_service  
    command_line      $USER1$/check_service.sh -s $ARG1$  
}
```

`command-name` : nom de la commande

`command_line` : syntaxe de la commande où `$USER1$` renvoie au répertoire libexec

`-s` correspond au service voulu qu'on va renseigner dans le prochain fichier de configuration.  
Ne pas oublier de sauvegarder.

Maintenant il faut créer un service nagios (quelques lignes qui définissent l'appareil à check, une description du service habituellement ce qui est check et la commande associée) qui va vérifier l'état d'un service voulu (service linux comme ssh, apache ou autre).

Dans le même repertoire, « `sudo nano localhost.cfg` »

```
define service {  
    use                local-service          ; Name of service template to use  
    host_name          localhost  
    service_description APACHE  
    check_command       check_service!apache2.service  
}
```

On copie les autres lignes et on remplace par ce qui nous intéresse.

J'ai choisi le service apache pour ma part.

Dans la ligne `check_command`, on entre le nom commande qu'on a créé tout à l'heure puis un « ! » qu sert à ajouter un argument (soit le premier -s on va juste mettre le nom du service voulu à la place)

On teste la configuration avec

**`sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg`** (on peut créer un alias dans  `~/.bashrc` et ajouter la ligne ;

**`alias testNagios = 'sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg'`**)

```
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
```

Pas d'erreurs.

On restart : **`sudo systemctl restart nagios.service`** et on va voir notre page nagios.

Après un petit temps ça fonctionne.

Host ▼	Service ▼	Status ▼	Last Check ▼	Duration ▼	Attempt ▼	Status Information
localhost	APACHE	OK	11-08-2022 09:21:46	0d 0h 9m 23s	1/4	Active: active (running) since Tue 2022-11-08 08:26:11 CET; 55min ago
	Current Load	OK	11-08-2022 09:22:00	7d 22h 46m 34s	1/4	OK - Charge moyenne: 0.00, 0.00, 0.00