

Table des matières

1. Installation et configuration de CentCTL.....	1
2. Test de la configuration.....	3
3. Automatiser le déploiement	3

1. Installation et configuration de CentCTL

Pour installer et configurer CentCTL nous nous aidons du site d'Aukfood :

<https://www.aukfood.fr/supervision-tour-dhorizon-avec-centctl-pour-centreon-open-source/>

/ ! \ L'installation se fait avec un utilisateur normal (pas en root)

Installation :

Nous devons commencer par créer un répertoire nommé centctl et nous installons l'outil dedans :

```
mkdir $HOME/centctl  
cd $HOME/centctl  
wget https://github.com/YPSI-SAS/centctl/releases/download/v2.5/centctl_linux_amd64 -O centctl
```

Il nous faut rendre le fichier exécutable :

```
chmod +x centctl
```

Il faut maintenant que nous modifions le fichier .bashrc (situé dans /home/\$USER/.bashrc) :

Attention à bien remplacer le \$USER par votre nom d'utilisateur

```
export CENTCTL_CONF=/home/$USER/centctl/centctl.yaml  
source /usr/share/bash-completion/bash_completion
```

Nous devons aussi installer le fichier bash-completion si ce n'est pas fait :

```
sudo apt install bash-completion
```

On vérifie aussi que la complétion se fasse :

```
source ~/.bashrc  
type _init_completion
```

La commande doit nous ressortir quelque chose commençant comme ça :

```
debian@debian10:~/centctl$ type _init_completion
_init_completion is a function
_init_completion ()
{
    local exclude= flag outx errx inx OPTIND=1;
    while getopts "n:e:o:i:s" flag "$@"; do
        case $flag in
            n)
                exclude+= $OPTARG
            ...
            ...
            ...
            ...
        ...
    ...
}
```

Nous ajoutons maintenant le script de complétion dans le répertoire /etc/bash_completion.d
(attention à remplacer le \$USER par l'utilisateur de la machine) :

```
sudo su
/home/$USER/centctl/centctl completion bash > /etc/bash_completion.d/centctl
```

Si le répertoire n'existe pas il faut le créer :

```
mkdir /etc/bash_completion.d
```

Nous repassons en utilisateur normal :

```
exit
```

Configuration de CentCTL:

Nous devons maintenant construire un fichier de configuration .yaml pour déclarer notre serveur
(attention il faut créer le fichier dans /home/\$USER/centctl/) :

```
debian@debian10:~/centctl$ nano centctl.yaml
```

Dans ce fichier nous mettons ce qui suit :

```
servers:
- server: "NOM DU SERVEUR DONNE LORS DE LA CREATION"
  url: "http://[IP_SERVER]/centreon"
  login: "LOGIN LORS DE LA CREATION DU COMPTE"
  password: "MOT DE PASSE DONNE LORS DE LA CREATION"
  version: "v1"
```

2. Test de la configuration

Maintenant que CentCTL est bien installé et configuré nous pouvons tester quelques commandes pour vérifier son bon fonctionnement :

Lister les hôtes sur le serveur :

```
debian@debian10:~/centctl$ ./centctl --server=centreon list host --output csv
ID,Name,Alias,Address,Activate
2,debian,Debian10,192.168.7.1,1
29,Windows10,,192.168.6.1,1
```

Nous retrouvons bien les hôtes et les IP qui sont sur notre serveur

Il existe plusieurs commandes que nous pouvons passer par exemple pour exporter ou importer une configuration. Nous pouvons les retrouver sur le site donné en début de rapport

3. Automatiser le déploiement

Dans cette partie nous devons automatiser le déploiement d'un nouveau service sur un ensemble d'hôtes. Pour se faire nous devons sur CentCTL :

- Créer une liste d'hôtes (en récupérant nos hôtes créés dans les précédents TP)
- Créer une nouvelle commande/check
- Créer des templates de services/hôtes grâce à CentCTL
- Créer un script de déploiement

Création d'une liste d'hôte :

Pour la création d'une liste d'hôtes nous allons simplement récupérer les hôtes créés dans les TP précédents grâce à la commande suivante :

```
debian@debian10:~/centctl$ ./centctl --server=centreon export host -r ^.* -f hosts.csv
```

Nous pouvons vérifier que le fichier est bien présent :

```
debian@debian10:~/centctl$ ls
centctl centctl.yaml hosts.csv
```

Création d'une nouvelle commande/check :

On peut créer une commande de check de la façon suivante :

```
debian@debian10:~/centctl$ ./centctl --server=centreon add command -l "\$ICMP$/check_ping -H\n\$HOSTADDRESS$ -w 3,60% -c 5,80% -p 5 -4" -n ChC_CHECK_PING -t check
```

Syntaxe de la commande shell :

- 1) -n, --name string To define the name of the command
- 2) -t, --type string To define the type of the command (check, notif, misc or discovery)
- 3) -l, --line string To define the line of the command

Il faut mettre un antislash devant toutes les Macro sinon le shell va l'interpréter comme une variable vide et dans centreon nous aurons une erreur

Nous pouvons vérifier dans notre serveur centreon que la commande a bien été créée :

| Modify a Command

Check

Command Name *	ChC_CHECK_PING
Command Type	<input type="radio"/> Notification <input checked="" type="radio"/> Check <input type="radio"/> Misc <input type="radio"/> Discovery
\$ICMP\$/check_ping -H \$HOSTADDRESS\$ -w 3,60% -c 5,80% -p 5 -4	

Création des templates de services :

Maintenant que la commande est créée nous pouvons créer un template de services que l'on va lier à notre commande :

```
debian@debian10:~/centctl$ ./centctl --server=centreon add template service -a  
SFT_ChC_CHECK_PING -n ST_ChC_CHECK_PING  
INFO: The STPL ST_ChC_CHECK_PING is created
```

Il nous faut aussi modifier les paramètres généraux :

```
debian@debian10:~/centctl$ ./centctl --server=centreon modify template service -n  
ST_ChC_CHECK_PING -p check_command -v ChC_CHECK_PING  
INFO: The parameter check_command of the STPL ST_ChC_CHECK_PING is modified  
  
debian@debian10:~/centctl$ ./centctl --server=centreon modify template service -n  
ST_ChC_CHECK_PING -p check_period -v 24x7  
INFO: The parameter check_period of the STPL ST_ChC_CHECK_PING is modified  
  
debian@debian10:~/centctl$ ./centctl --server=centreon modify template service -n  
ST_ChC_CHECK_PING -p max_check_attempts -v 3  
INFO: The parameter max_check_attempts of the STPL ST_ChC_CHECK_PING is modified
```

-p : paramètre à modifier

-v : valeur que l'on veut mettre au paramètre

Nous pouvons vérifier que le template est bien crée dans centreon :

| Modify a Service Template Model

General Information

Alias * SFT_ChC_CHECK_PING

Name * ST_ChC_CHECK_PING

Template Template

Service Check Options

Check Command ChC_CHECK_PING

Custom macros

Template inheritance
Command inheritance

+ Add a new entry
Nothing here, use the "Add" button

Args

Argument	Value
No argument found for this command	

Service Scheduling Options

Check Period 24x7

Max Check Attempts 3

Normal Check Interval * 60 seconds

Retry Check Interval * 60 seconds

Active Checks Enabled ☐ Yes ☐ No ☒ Default

Passive Checks Enabled ☐ Yes ☐ No ☒ Default

Is volatile ☐ Yes ☐ No ☒ Default

Création des templates d'hôtes :

Maintenant que le template de services est créé nous pouvons créer un template d'hôtes qui va regrouper les différents services créés et qui sera lié à notre hôte (pour le moment nous n'avons qu'un seul service à des fins de test) :

```
debian@debian10:~/centctl$ ./centctl --server=centreon add template host -a  
HFT_DEBIAN_WEB_SERVER -n HT_DEBIAN_WEB_SERVER  
INFO: The HTPL HT_DEBIAN_WEB_SERVER is created
```

Maintenant que le template est créé il faut le lier à tous les templates de services que nous avons créés et avons besoin pour un hôte. Ici nous n'avons créé que le ping mais si nous avions plusieurs services pour un hôte Linux par exemple il faudrait lier tous les templates de services Linux au template d'hôte et donc répéter la commande suivante pour chaque service en modifiant la syntaxe du -v :

```
debian@debian10:~/centctl$ ./centctl --server=centreon modify template host -n  
HT_DEBIAN_WEB_SERVER -p linkedservice -v ST_ChC_CHECK_PING  
INFO: The parameter linkedservice of the STPL HT_DEBIAN_WEB_SERVER is modified
```

Nous pouvons vérifier dans centreon que le template de service ping est bien lié à notre service d'hôte :

| [Modify relations](#)

Hostgroup Relations

Linked Service Templates

ST_ChC_CHECK_PING

Création d'un hôte :

Il ne nous reste plus qu'à créer un hôte avec un nom et une IP puis le lier à notre template d'hôte qui regroupe les différents services (nous pourrions au lieu de créer un hôte, lier le template à un hôte déjà existant) :

```
debian@debian10:~/centctl$ ./centctl --server=centreon add host -i 192.168.7.1 -a debian_test_ping -n
debian_test_ping -p Central -t HT_DEBIAN_WEB_SERVER
INFO: The host debian_test_ping is created
INFO: The template of the host debian_test_ping is applied
```

-i : IP de l'hôte

-a : Alias au nom de l'hôte

-n : Nom de l'hôte

-p : Serveur qui supervise cet hôte

-t : Template à appliquer sur l'hôte

Nous pouvons vérifier dans centreon que l'hôte est bien créé et lié au template d'hôte :

| [Modify a Host](#)

Host basic information

Name * debian_test_ping

Alias debian_test_ping

IP Address / DNS * 192.168.7.1 [Resolve](#)

SNMP Community & Version

Monitored from Central

Timezone / Location

Templates

A host can have multiple templates, their orders have a significant importance
[Here is a self-explanatory image.](#)

HT_DEBIAN_WEB_SERVER

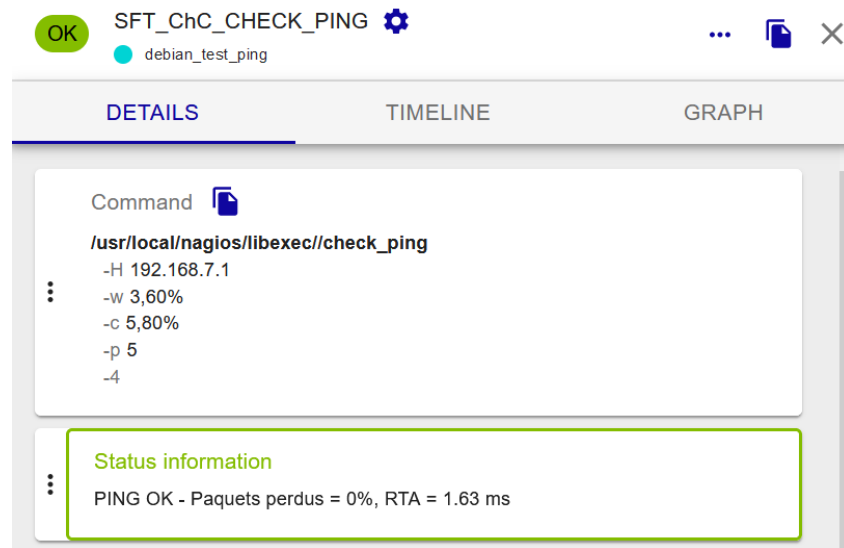
Create Services linked to the Template too ☐ Yes ☒ No

Appliquer les modifications sur le serveur :

Pour appliquer les modifications sur le serveur il faut exporter le fichier de configuration de la façon suivante :

```
debian@debian10:~/centctl$ ./centctl --server=centreon apply -n Central
INFO: The configuration of the poller Central is exported
```

Nous pouvons maintenant vérifier dans l'Onglet Monitoring que notre hôte est bien supervisé au niveau du service ping :



Création d'un script de déploiement :

Pour créer un script de déploiement nous devons commencer par récupérer les hôtes de notre serveur et ne garder que leur nom :

```
debian@debian10:~/centctl$ ./centctl --server=centreon list host --output csv |cut -d, -f 2 > hosts.csv
```

Le cut permet de ne garder que le 2^{ème} champ de notre fichier ce qui correspond au nom des hôtes

Le "> hosts.csv" permet d'écrire le résultat de la commande dans le fichier hosts.csv et d'écraser les anciennes données s'il y en avait

Il faut éditer le fichier hosts.csv et supprimer la 1^{ère} ligne avec "Name"

Maintenant nous avons notre fichier hosts de la façon suivante :

```
Windows10
debian
```

Nous pouvons donc maintenant créer un script qui va récupérer le nom des hôtes dans le fichier hosts.csv et qui va créer différentes commandes de supervision en fonction de l'hôte sélectionné

Ici dans notre script nous avons créé 3 commandes : 1 en NRPE, 1 en SSH et 1 en SNMP

Le script va créer pour chaque hôte une commande, un template de services, un template d'hôte puis va lier le template d'hôte à l'hôte

Pour terminer le script va exporter la configuration dans le serveur principal afin de mettre à jour les données

Le script est donc le suivant :

```
#!/bin/bash

#Déclare la liste d'hôte
listhost="$(cat hosts.csv)"

#Boucle sur tout les hôtes
for host in $listhost; do

    #Création d'une supervision NRPE sur la machine Windows
    if [ "$host" == "Windows10" ]; then

        echo "Création de la commande NRPE pour l'hôte $host : "

        ./centctl --server=centreon add command -l "\$USER1$/check_centreon_nrpe3 -H
        \$HOSTADDRESS$ -c check_updates" -n check_updates_nrpe

        echo "Commande NRPE créée"

        echo "Création d'un template de service pour le NRPE"

        ./centctl --server=centreon add template service -a updates_nrpe -n updates_nrpe

        ./centctl --server=centreon modify template service -n updates_nrpe -p check_command -v
        check_updates_nrpe

        ./centctl --server=centreon modify template service -n updates_nrpe -p check_period -v
        24x7

        ./centctl --server=centreon modify template service -n updates_nrpe -p
        max_check_attempts -v 3

        echo "Template de services NRPE créée"

        echo "Création d'un template d'hôtes"

        ./centctl --server=centreon add template host -a template_windows_nrpe -n
        template_windows_nrpe

        ./centctl --server=centreon modify template host -n template_windows_nrpe -p
        linkedservice -v updates_nrpe

        echo "Template d'hôte créée"

        echo "Liaison du template sur l'hôte"

        ./centctl --server=centreon modify host -n $host -p template -v template_windows_nrpe

    fi

#Création d'une supervision by SSH sur la machine debian
    if [ "$host" == "debian" ]; then

        echo "Création de la commande by SSH pour l'hôte $host : "
```



```
./centctl --server=centreon add command -l "\$LOCAL$/check_by_ssh -H  
|\$HOSTADDRESS$ -i ""\$SSH$"" -l \$HOSTNAME$ -C \$LOCAL$/che$  
  
echo "Commande SSH créée"  
  
echo "Création d'un template de service pour le SSH"  
  
./centctl --server=centreon add template service -a check_ssh_apache -n  
check_ssh_apache  
  
./centctl --server=centreon modify template service -n check_ssh_apache -p  
check_command -v check_by_ssh  
  
./centctl --server=centreon modify template service -n check_ssh_apache -p check_period  
-v 24x7  
  
./centctl --server=centreon modify template service -n check_ssh_apache -p  
max_check_attempts -v 3  
  
echo "Template de services SSH créée"  
  
echo "Création d'un template d'hôtes"  
  
./centctl --server=centreon add template host -a template_debian -n template_debian  
  
./centctl --server=centreon modify template host -n template_debian -p linkedservice -v  
check_ssh_apache  
  
echo "Template d'hôte créée"  
  
#Création d'une supervision SNMP sur la machine debian  
  
echo "Création de la commande SNMP pour l'hôte $host :"  
  
./centctl --server=centreon add command -l  
"\$CENTREONPLUGINS$/centreon_linux_snmp.pl --plugin=os::linux::snmp::plugin --mode=cpu$  
  
echo "Commande SNMP créée"  
  
echo "Création d'un template de service pour le SNMP"  
  
./centctl --server=centreon add template service -a check_snmp -n check_snmp  
  
./centctl --server=centreon modify template service -n check_snmp -p check_command -v  
check_cpu_snmp  
  
./centctl --server=centreon modify template service -n check_snmp -p check_period -v  
24x7  
  
./centctl --server=centreon modify template service -n check_snmp -p  
max_check_attempts -v 3  
  
echo "Template de services SNMP créée"  
  
echo "Association au template actuel pour l'hôte"  
  
./centctl --server=centreon modify template host -n template_debian -p linkedservice -v  
check_snmp  
  
echo "Association terminée"  
  
echo "Liaison du template sur l'hôte"
```

```
./centctl --server=centreon modify host -n $host -p template -v template_debian  
fi  
  
done  
#Exporte la configuration sur le poller selectionné et le recharge  
./centctl --server=centreon apply -n Central  
  
echo "Le déploiement de la supervision est terminé"  
echo "Vous pouvez ouvrir le serveur dans un navigateur web à l'adresse suivante : "  
echo "http://IP_Serveur/centreon"  
echo "Vérifiez le bon fonctionnement du script en regardant l'onglet Monitoring"
```

Nous pouvons donc lancer le script :

```
debian@debian10:~/centctl$ ./autodeploy.sh  
Création de la commande NRPE pour l'hôte Windows10 :  
INFO: The CMD check_updates_nrpe is created  
Commande NRPE créée  
Création d'un template de service pour le NRPE  
INFO: The STPL updates_nrpe is created  
INFO: The parameter check_command of the STPL updates_nrpe is modified  
INFO: The parameter check_period of the STPL updates_nrpe is modified  
INFO: The parameter max_check_attempts of the STPL updates_nrpe is modified  
Template de services NRPE créée  
Création d'un template d'hôtes  
INFO: The HTPL template_windows_nrpe is created  
INFO: The parameter linkedservice of the STPL template_windows_nrpe is modified  
Template d'hôte créée  
Liaison du template sur l'hôte  
INFO: The parameter template of the host Windows10 is modified  
INFO: The template of the host Windows10 is applied  
Création de la commande by SSH pour l'hôte debian :  
INFO: The CMD check_by_ssh is created  
Commande SSH créée  
Création d'un template de service pour le SSH
```

INFO: The STPL check_ssh_apache is created

INFO: The parameter check_command of the STPL check_ssh_apache is modified

INFO: The parameter check_period of the STPL check_ssh_apache is modified

INFO: The parameter max_check_attempts of the STPL check_ssh_apache is modified

Template de services SSH créée

Création d'un template d'hôtes

INFO: The HTPL template_debian is created

INFO: The parameter linkedservice of the STPL template_debian is modified

Template d'hôte créée

Création de la commande SNMP pour l'hôte debian :

INFO: The CMD check_cpu_snmp is created

Commande SNMP créée

Création d'un template de service pour le SNMP

INFO: The STPL check_snmp is created

INFO: The parameter check_command of the STPL check_snmp is modified

INFO: The parameter check_period of the STPL check_snmp is modified

INFO: The parameter max_check_attempts of the STPL check_snmp is modified

Template de services SNMP créée

Association au template actuel pour l'hôte

INFO: The parameter linkedservice of the STPL template_debian is modified

Association terminée

Liaison du template sur l'hôte

INFO: The parameter template of the host debian is modified

INFO: The template of the host debian is applied

INFO: The configuration of the poller Central is exported

Le déploiement de la supervision est terminé

Vous pouvez ouvrir le serveur dans un navigateur web à l'adresse suivante :

http://IP_Serveur/centreon

Vérifiez le bon fonctionnement du script en regardant l'onglet Monitoring

Notre script a bien fonctionné et nous pouvons voir dans l'onglet "Monitoring que nos services sont mis en place :

Pour l'hôte debian :

UNKNOWN	check_snmp UNKNOWN: SNMP Table Request: Cant get a single value.	1h 37m
OK	check_ssh_apache PROCS OK: 6 processes with args 'apache2'	1h 37m

La commande de check SNMP est fonctionnelle sur le serveur mais comme le SNMP n'est pas configuré sur ma machine le serveur nous remonte un statut « UNKNOWN »

Pour l'hôte Windows10 :

CRITICAL	updates_nrpe CHECK_NRPE: Invalid packet version received from server. Updates: 1 critical, 0 optional Mise à jour intelligente de la sécurité, pour Microsoft Defender Antivirus - KB2267602 (ver...	1h 37m
----------	---	--------

Nous aurions pu mettre des arguments dans nos commandes mais nous n'avons pas trouvé la commande pour définir un argument dans le template de service

Création d'une commande avec argument (attention aux antislash) :

```
debian@debian10:~/centctl$ ./centctl --server=centreon add command -l  
"$USER1$/check_centreon_nrpe3 -H $HOSTADDRESS$ -c check_cpu -a filter=none 'warn=load >  
|$ARG1$' 'crit=load > |$ARG2$'" -n cpu_nrpe -t check  
INFO: The CMD cpu_nrpe is created
```

Modification d'une commande avec argument (attention aux antislash) :

```
debian@debian10:~/centctl$ ./centctl --server=centreon modify command -p command_line -v  
"$USER1$/check_centreon_nrpe3 -H $HOSTADDRESS$ -c check_cpu -a filter=none 'warn=load >  
|$ARG1$' 'crit=load > |$ARG2$'" -n cpu_nrpe  
INFO: The parameter command_line of the CMD cpu_nrpe is modified
```