
Installation et exploitation Centreon Documentation

Version 1.0.0

Aurelazy

nov. 13, 2017

Table des matières

1	Installation	3
1.1	Client NTP	3
2	Les Hôtes	5
2.1	Création d'un hôte	6
2.2	Création d'un modèle d'hôte	6
2.3	Groupes d'hôtes	7
3	Les services	9
3.1	Les services par hôte	9
3.2	Les services par groupe d'hôtes	9
3.3	Création d'un modèle de service	10
4	Les graphiques	11
5	Les utilisateurs	13
6	Les ACLs	15
6.1	La gestion des groupes d'accès	15
6.2	Gestion des accès aux menus	16
6.3	Gestion des accès aux ressources	16
6.4	Gestion des accès sur les actions	16
7	Le moteur de supervision	17
8	Les commandes	19
8.1	check_traffic	20
9	Les scripts	21
10	Sauvegardes	23
10.1	Sauvegarde des fichiers de configuration Centreon	23
10.2	Sauvegarde des BDD MySQL	24
10.3	Récupération des sauvegardes	26
11	Création utilisateur MySQL	29
12	Sauvegarde configuration routeurs	31

12.1	Création d'un fichier host	31
12.2	Ajout d'un nouvel hôte	32
12.3	Ajout des paquets utiles au script	33
12.4	Création d'un script	33
12.5	Lancement du script	35
13	Installation de NagVis	37
13.1	Installation du broker MK Livestatus	37
13.2	Configuration de l'interface Centreon	38
13.3	Installation de Graphviz	38
13.4	Installation de NagVis	39
14	Ce qu'il reste à faire	41
15	Indices and tables	43

Contents :

CHAPITRE 1

Installation

Nous n'allons pas expliquer pas à pas l'installation de Centreon (leur [site](#) le fait très bien). Nous allons voir les ajouts que nous avons fait, les changements...

Nous verrons dans les prochains chapitres :

- *La création d'un hôte*
- *La création d'un service*
- *Les graphiques*
- Les plugins Les plugins
- Etc ...

Sur le serveur le compte par défaut est `root`. Nous allons ajouter un autre utilisateur. Nous arrons donc 2 utilisateurs :

- `root`
- `aurelazy` # Bien sûr vous le nommer comme vous voulez

La BDD contient déjà 2 utilisateurs, `root`, `centreon`. Pour notre projet, nous créerons 1 utilisateur en plus pour les sauvegardes, nous le nommerons `backup`.

Donc voici nos différents comptes pour MySQL :

- `root` # Attention à bien ajouter un mot de passe, car vide
- `centreon`
- `backup`

1.1 Client NTP

Sur le serveur, nous allons ajouter un client NTP pour que notre serveur soit à jour

Il faudra éditer le fichier `/etc/ntp.conf` :

```
# vim /etc/ntp.conf
```

Insérez le(s) nom(s) ou adresse(s) de votre serveur de temps dans ce fichier :

```
# undisciplined local clock. This is a fake driver intended for backup and when no
↪outside source of
# synchronized time is available
```

```
server 127.127.1.0 # local clock
fudge 127.127.1.0 stratum 10
server <TIMESERVERADORE>
server <TIMESERVERADORE>
```

Redémarrer le service NTP :

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

Vérifiez la date du serveur :

```
date
```


CHAPITRE 2

Les Hôtes

Pour notre projet, un hôte sera un routeur, switch, . . . , (le process sera le même pour tout autre équipement, comme un serveur, un PC, ...) les éléments indispensables pour la création de celui-ci sont :

- Un nom
- Une adresse IP
- Son modèle
- Pour d’autres projets, on aura besoin du type de l’équipement (serveur Linux, Windows, tout équipement IP)

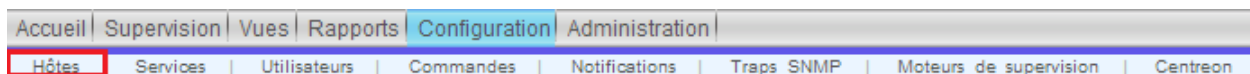
La description de l’hôte sera son emplacement (ex : Paris), cette information est utile mais pas indispensable. Le nom de l’hôte est sous la forme “RT_20nn”, n est un nombre, on utilisera l’adresse IP pour remplacer les “n”, ex : @ IP ==> 10.240.2.21 ==> RT_2021

Le modèle de celui-ci est très important également. Il servira à lui assigner les bons services.

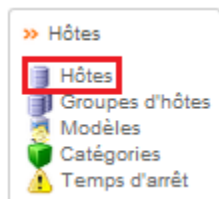
Sur le réseau nous avons 4 modèles (les noms des routeurs seront imaginés) de routeurs différents :

- **CISCO ZE-P2549-**
 - 48
 - 24
- CISCO ZE-P2549G-24
- JUNIPER SRX24
- JUNIPER MX5

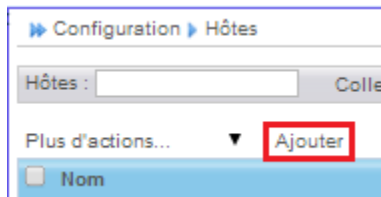
Pour la configuration, l’ajout d’un hôte, il suffit de cliquer sur Configuration ==> Hôtes



2.1 Création d'un hôte



Pour la création d'un hôte, il suffit de cliquer sur *Ajouter*



Disons que nous devons créer un nouvel équipement “SW_2048” qui se trouve à Marseille, il s’agit d’un routeur Juniper MX5

Voici comment remplir le “formulaire” :

- Le nom de l’hôte : “SW_2048”
- Alias : “Marseille”
- Adresse IP : 10.240.2.48
- Modèle d’hôte : “Routeur-Juniper-MX”
- Créer ... : “Oui”

Dans l’onglet “Relations” :

- *Groupe d’hôtes* parents : “Routers”

Puis Sauvegarder.

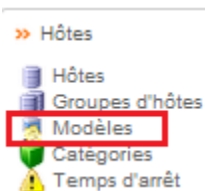
L’option qui va permettre d’ajouter automatiquement *les services* à notre hôte est “*Modèle d’hôte*”, il faut cocher également “Créer aussi les services liés au modèle” pour qu’ils se greffent à notre hôte.

2.2 Création d'un modèle d'hôte

Un modèle d’hôte est un objet de supervision Centreon qui possède toutes les caractéristiques définissant un hôte et pouvant être utilisé comme base pour la configuration d’un hôte. Un modèle d’hôte est à l’hôte ce qu’un modèle de service est à un service : un objet préconfiguré pour être utilisé dans des hôtes. Les caractéristiques sont les mêmes :

- héritage : un hôte peut hériter d’un modèle d’hôte et un modèle d’hôte peut hériter d’un autre modèle d’hôte parent.
- surcharge : un hôte héritant d’un modèle d’hôte peut modifier (« surcharger ») le paramètre hérité par le modèle.

Nous allons voir comment est construit le modèle d’hôte “Routeur-Juniper-MX”.



Il faut cliquer sur “Ajouter”. Voici les informations à remplir dans le premier onglet “Configuration de l’hôte” :

- Nom du modèle d’hôte : “Routeur-Juniper-MX”

- Alias : “Juniper MX Routers”
- Communauté SNMP & Version” : “<communauté & version>”
- Modèles d’hôte parallèles : “generic-host”

Nous mettons en héritage “generic-host” créé par défaut, qui permet de récupérer le service ping.

Puis nous passons à l’onglet “Relations”, dans cet onglet, nous ajoutons les “Modèles de service liés”.

Nous utiliserons pour ce modèle d’hôte les *modèles de services* suivant :

- Traffic_JUNIPER_GigabitEthernet-1/0/[0-9]
- Traffic_JUNIPER_GigabitEthernet-1/1/[0-9]

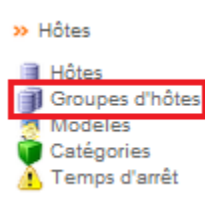
[0-9] veut dire de 0 à 9

Puis, on sauvegarde.

Pour notre projet, nous avons créé 4 modèles d’hôtes :

- Router-Cisco : Pour les modèles de routeurs “CISCO ZE-P2549-48/24”, il comprend 52 services (48 fastEthernet + 4 GigabitEthernet).
- Router-Cisco_G : Pour les modèles de routeurs ” CISCO ZE-P2549G-24”, il comprend 24 services (24 GigabitEthernet).
- Routeur-Juniper-MX : Pour les modèles de routeurs ” Juniper MX”, il comprend 20 services (20 GigabitEthernet).
- Router-Juniper-SRX : Pour les modèles de routeurs “Juniper SRX24”, il comprend 16 services (16 GigabitEthernet).

2.3 Groupes d’hôtes

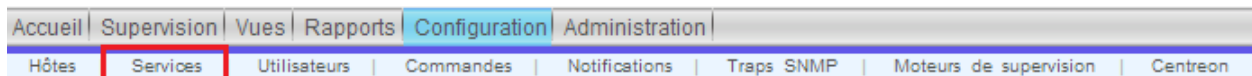


Nous utiliserons un groupe d’hôtes prédéfinie lors de l’installation de Centreon, “Routers”.

CHAPITRE 3

Les services

Les services sont des indicateurs que nous mettons sur nos hôtes pour vérifier si tel ou tel fonction de notre hôte fonctionne. Le service le plus commun et utilisé sur pratiquement tous les équipements est le “ping”. Ce service va nous permettre de vérifier que l’équipement répond. Nous pouvons choisir le temps entre 2 “ping”, entre autre. Nous verrons plus loin comment créer ceux-ci.



3.1 Les services par hôte

Cette configuration n’est pas utilisée dans notre projet ! Ceci permet d’implanter un service à un hôte et seulement un hôte, il est préférable d’utiliser *les services par groupes d’hôte* ou *les modèles de service*.

Nous utiliserons ce menu pour désactiver certains services.



3.2 Les services par groupe d’hôtes

Cette configuration n’est pas utilisée dans notre projet !

Note : Nous verrons plus tard pour ajouter des informations sur cette configuration

3.3 Création d'un modèle de service

Nous allons voir comment créer un modèle de service.



Nous voulons superviser les ports sur les routeurs/commutateurs de notre réseau, le plugin utilisé est `/usr/lib/nagios/plugins/check_iftraffic3.pl` qui remonte la bande passante du port, ainsi que le statut administratif/opérationnel de celui-ci (le plugin a été modifié pour remonter cette dernière information). Nous mettons en relation ce plugin avec la *commande* `check_traffic`.

Nous créons un modèle de service "générique" pour le trafic, "Traffic_CISCO" (il sera utilisé aussi pour les Juniper).

Nous ajoutons donc un modèle en cliquant sur "Ajouter". Voici les informations que nous avons rentré dans le premier onglet, "Configuration du service" :

- Alias : "Traffic_CISCO"
- Nom du modèle de service : "Traffic_CISCO"
- Modèle de service : "generic-service"
- Est volatile : "Non"
- Période de contrôle : "24x7"
- Commande de vérification : "check_traffic"
- Nombre de contrôle avant validation de l'état : "5"
- Intervalle normal de contrôle : "5"
- Intervalle non-régulier de contrôle : "1"

Le reste est par défaut.

Dans l'onglet "Informations supplémentaires du service", dans "*Modèle de graphique*", on ajoute "Traffic".

On peut sauvegarder.

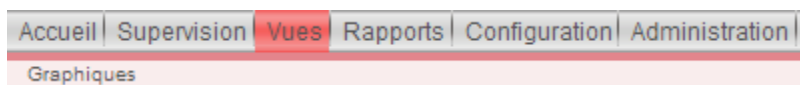
Puis l'on crée des modèles de service moins générique qui vont hériter de "Traffic_CISCO". Nous allons avoir 4 "familles" de modèles suivant le nom donnés par les routeur/commutateurs pour les ports.

- Traffic_Cisco_FastEthernet-1/0/[1-48]
- Traffic_Cisco_GigabitEthernet-1/0/[1-24]
- Traffic_JUNIPER_GigabitEthernet-1/[0-1]/[0-9]
- Traffic_JUNIPER_GigabitEthernet-0/0/[0-15]

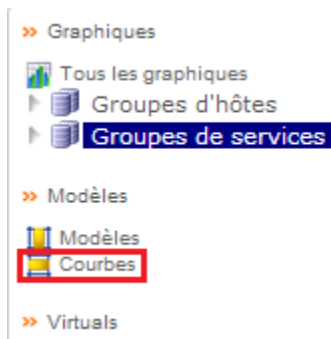
CHAPITRE 4

Les graphiques

Comme son nom l'indique, les graphiques permettent de visualiser l'historique d'état d'un service.



Pour les services “Traffic_*”, nous avons ajouté une “courbe” ou indicateur pour récupérer l'historique de la bande passante des ports.



Nous avons créé les courbes “inBandwidth” et “outBandwidth”. Voici les informations :

Tableau 4.1 – Info Graph

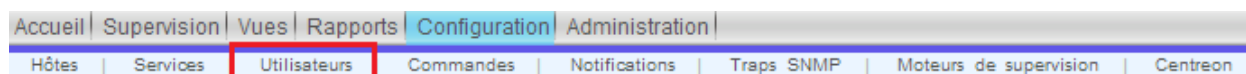
Nom du modèle	inBandwidth	outBandwidth
Nom de la source de données	inBAndwidth	outBandwidth
Ordre	1	1
Inverser	Non	Oui
Couleur de la courbe (Manuel)	#1EE045	#FF0000
Couleur de l'aire	#1EE045	#FF0000
Transparence	80%	80%
Remplissage	Oui	Oui
Afficher la valeur maximale	Oui	Oui
Afficher la moyenne	Oui	Oui
Afficher la dernière valeur	Oui	Oui

Toutes les autres options seront par défaut.

CHAPITRE 5

Les utilisateurs

Il est conseillé de créer un compte qui aura moins de droits que le compte `admin` par défaut et de créer des droits d'accès (ACL) spécifiques. Se qui permettra de laisser la supervision à un groupe des personnes qui auront seulement le droit de voir les alarmes et au cas où les acquitter.

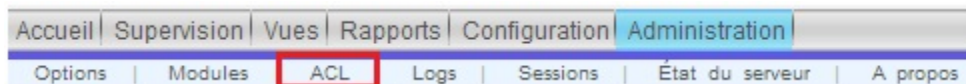


Pour la création il suffit de suivre chaque option et les remplir. Si vous avez suivi l'installation depuis le site de Centreon, vous aurez la possibilité de choisir la langue française.

CHAPITRE 6

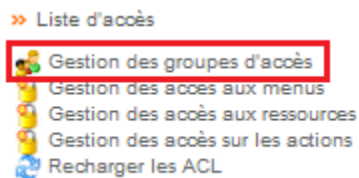
Les ACLs

Les ACL (Access Control List) permettent de limiter l'accès à des ressources, menus et/ou actions, nous allons voir dans ce chapitre comment les mettre en place. Nous prenons pour exemple une région qui aura le droit d'accéder uniquement à ses équipements, par exemple : la région de Paris, PARIS. Dans le menu centreon, cliquer sur Administration ==> ACL



6.1 La gestion des groupes d'accès

Nous allons commencer par le premier menu, "Gestion des groupes d'accès" :



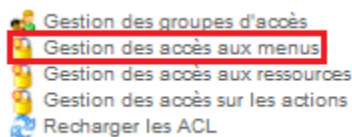
Voici les principales informations :

- Nom du groupe : "Paris_tech"
- Alias : "Paris_tech"
- *Contacts liés* : "Tech_Paris"

Pour l'onglet "Autorisations" :

- *Accès aux ressources* : "Paris_ressources"
- *Accès aux menus* : "Menu_Sup"
- *Actions autorisées* : "Simple User"

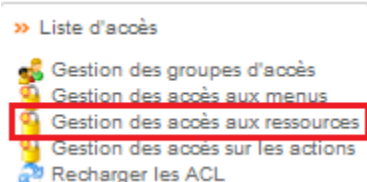
6.2 Gestion des accès aux menus



Voici les accès aux menus que nous avons configurés pour les techniciens de Paris :

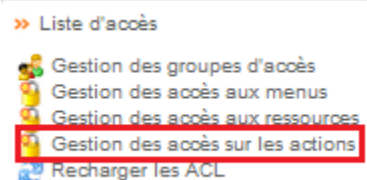
- Les pages disponibles : Accueil & Supervision

6.3 Gestion des accès aux ressources



Pour cet accès, nous avons inclus tous les hôtes excepté le serveur Centreon, “Centreon-Server”. Et inclus tous les services.

6.4 Gestion des accès sur les actions



Nous avons, pour les techniciens, utilisé le groupe “Simple User”. Voici ce qu’ils ont droit de faire : Dans le menu “Accès aux fonctionnalités globales”, ils ont vu sur tout sauf “Afficher la liste des collecteurs (pas de collecteur dans l’architecture).

Dans le menu “Accès aux actions sur les services” :

- Acknowledge a service (Acquitter un service)
- Disaknowledge a service
- Re-planifier la prochaine vérification d’un service
- Re-planifier la prochaine vérification d’un service (Forcé)
- Ajouter/Supprimer un commentaire pour un service

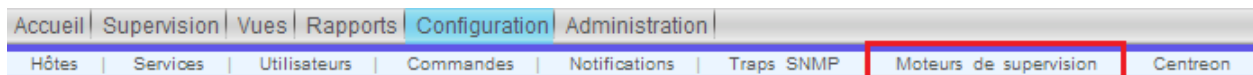
Dans le menu “Accès aux actions sur les hôtes” :

- Acquitter un hôte
- Disaknowledge a host
- Planifier la prochaine vérification d’un hôte
- Planifier la prochaine vérification d’un hôte (Forcé)
- Ajouter/Supprimer un commentaire pour un hôte

CHAPITRE 7

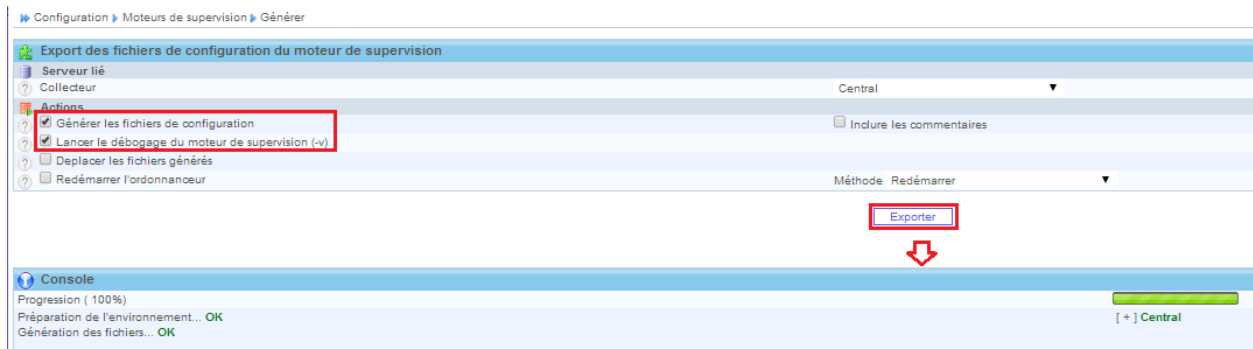
Le moteur de supervision

Après un changement dans la configuration de Centreon (ajout/suppression d'hôtes, de services, ou des modifications) il est conseillé de générer les fichiers de configuration et de redémarrer l'ordonnanceur. Voici la marche à suivre :

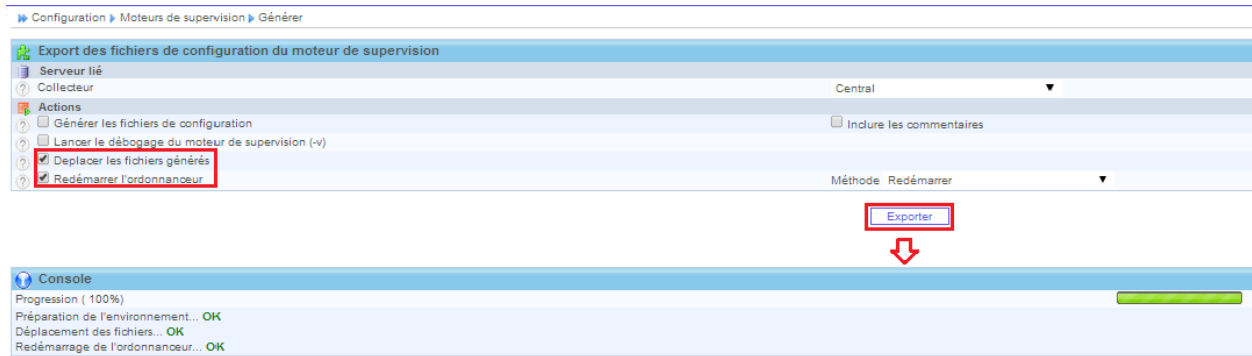


Ceci se fait en 2 parties :

— 1ere partie :



On vérifie qu'aucune erreur n'apparaît lors de la génération des fichiers de configuration. Si aucune erreur, on passe à la partie 2, sinon vérifier l'erreur et régler celle-ci et recommencer depuis la partie 1 pour vérifier que tout a été arrangé. 2eme partie :



Normalement, si la première partie n'a soulevé aucun problème, cette partie sera en état "OK".

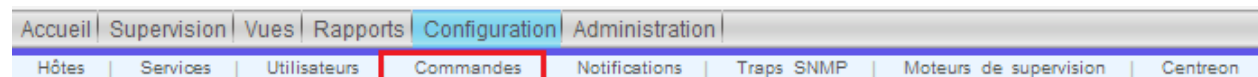
Note : Il peut arriver qu'un warning apparaisse en disant que le groupe `Guest` ne contient pas d'utilisateur. Ceci n'est pas bloquant, il suffit de rajouter un utilisateur ou de désactiver celui-ci.

CHAPITRE 8

Les commandes

Les commandes utilisent des plugins (scripts) pour valider l'état d'un équipement, comme le ping ou la charge CPU. Elles fonctionnent un peu comme les commandes shell, c'est-à-dire, qu'elles prennent des options ou arguments, comme l'adresse IP de l'équipement, la communauté, version snmp, ...

C'est un des point les plus important dans Centreon (Nagios). Ce sont elles qui interrogerons les équipements par le biais d'un service.



Les commandes sont utilisées dans *les services* ou les event handlers. Dans celles-ci, nous utilisons des “macros”, c'est une option qui va changer de valeur suivant le contexte. Nous avons par exemple une macro “\$HOSTADDRESS\$”, qui va récupérer l'adresse IP de l'équipement pour qui on va lancer la commande.

The screenshot shows the 'Ajouter une commande' (Add command) form in the Centreon interface. The form is divided into several sections. The top section is 'Vérification' (Verification), which includes fields for 'Nom de commande' (Command name), 'Type de commande' (Command type), and 'Ligne de commande' (Command line). The 'Type de commande' field is set to 'Vérification'. The 'Ligne de commande' field contains a macro '\$HOSTADDRESS\$'. To the right of the 'Ligne de commande' field, there are three dropdown menus: '\$USER1\$ (path to the plugins)', '/Aurel/. Trap_ACMi.py.swp', and '\$ADMINEMAILS\$'. The bottom section is 'Informations supplémentaires' (Additional information), which includes fields for 'Connecteurs' (Connectors) and 'Modèle de graphique' (Graph model). The form is numbered 1 through 8, indicating the sequence of steps to complete the command setup.

Voici les principales informations pour la création d'une commande.

1. Le nom de la commande, utilisé dans *les services*.
2. Dans la plupart des cas le type de commande sera "Vérification"
3. Dans cette partie nous indiquons le plugin utilisé avec ces options éventuelles.
4. Ici nous trouvons une macro "\$USER1\$" qui donne le chemin par défaut des plugins de Nagios/Centeron ("/usr/lib/nagios/plugins")
5. Ici nous indiquons le plugin que nous voulons utiliser.
6. Dans ce menu, nous voyons les macros que nous pouvons utiliser, comme \$HOSTADDRESS\$, \$_HOSTSNMPCOMMUNITY\$, \$ARGn\$ utiliser pour des options qui seront à changer si nécessaire, comme les seuils (le n est à changer par un nombre)
7. Ici, nous indiquons un seuil ou option par défaut, surtout lors d'utilisation de la macro \$ARGn\$.
8. Ici nous décrivons les arguments qui utilisent les macros \$ARGn\$.

Nous verrons en détail comment remplir tous ces champs par la suite lorsque nous expliquerons les commandes personnelles.

8.1 check_traffic

Cette commande va permettre de récupérer la bande passante de chacun de nos ports sur chacun de nos routeurs/commutateurs, ainsi que son état administratif (up/down) et opérationnel (up/down). Par exemple, si notre port est dans l'état opérationnel est UP et que l'état administratif est DOWN, le service retournera un état CRITIQUE idem s'il dépasse les seuils définis pour la bande passante. Dans ce projet, nous voulons être au courant lorsqu'un port est ouvert mais que rien n'est racordé dessus

Le plugin utilisé est "/usr/lib/nagios/plugins/check_iftraffic3.pl" un peu remanié par rapport à l'original. Nous verrons cela dans la section "*scripts*".

Les options passées au plugin sont :

- -H \$HOSTADDRESS\$
- -C \$_HOSTSNMPCOMMUNITY\$ ==> la communauté SNMP de l'équipement
- -i \$ARG1\$ ==> le port à vérifier (ex : ge-1/1/0)
- **-B** ==> récupère le résultat en bits

CHAPITRE 9

Les scripts

ici les explications des changements fait sur le plugin “/usr/lib/nagios/plugins/check_iftraffic3.pl”.

Le script ne prenait pas en compte l'état administratif du port, c'est-à-dire que si le port était ouvert aucune alarme n'était remontée. Rajout également du nom du lien dans la sortie détaillée. Ligne 99, ajout de la variable “\$snmpIfAdminStatus = ‘1.3.6.1.2.1.2.2.1.7’” Ligne 107, ajout de la variable “\$snmpDescrLien = ‘1.3.6.1.2.1.31.1.1.1.18’” Ligne 118, ajout de la variable “if_status_admin”, qui va permettre de récupérer le résultat de la commande snmp sur l’OID précédente, à la ligne 279. Lignes 286-296, ajout d’une comparaison du résultat récupéré avec le code retour pour UP (1), si celui-ci n’est pas égal à 1 alors le service sera en OK et on sort du script car le port sera fermé DOWN (2) ! Ligne 289, ajout de la variable “\$Descr_Lien”, qui va récupérer le nom du lien, avec la ligne 260.

Changement dans le retour des données de performance du plugin. Les données “inAbsolut” et “outAbsolut” sortaient des graphiques incompréhensibles et non lisibles.

J’ai donc enlevé ces informations du plugin, lignes 287, 425.

NB : Si vous avez besoin du script n’hésitez pas à me demander depuis le compte [GitHub](#).

On va sauvegarder les fichiers important de Centreon, ainsi que toutes les BDD tous les soirs grâce à 2 scripts de sauvegarde qui seront lancés par une tâche cron. Les scripts se trouvent dans le dossier “/home/<user>/script_sauvegarde/”. Le premier, celui qui sauvegarde la configuration ainsi que les fichiers/scripts/plugins . . . , se nomme “sauvegarde.sh”, le second qui sauvegardera nos BDD Centreon se nomme “sauv_mysql.sh”.

10.1 Sauvegarde des fichiers de configuration Centreon

Voici le script sauvegarde.sh.

```
#!/bin/bash

DATE=`date +%y%m%d`
rep=/var/spool/centreon/dump/
dest=/home/backup/sauvegardes/dump_centreon/dump_centreon.${DATE}.tar.gz

# Création du répertoire à compresser
mkdir ${rep}
# Suppression de l'ancienne sauvegarde
#rm -f ${dest}

mkdir ${rep}httpdconf/
mkdir ${rep}nagioskeys/
mkdir ${rep}rootkeys/

NbArchive=$(ls -A /home/backup/sauvegardes/dump-centreon/ | wc -l)
if [${NbArchive} -gt 7 ]
then
    # On recupere l'archive la plus ancienne
    Old_backup=$(ls -lrt /home/backup/sauvegardes/dump_centreon/ |grep ".tar.g" |
    ↪|head -n 1 |cut -d":" -f 2 |cut -d " " -f 2)
    rm -f /home/backup/sauvegardes/dump_centreon/${Old_backup}
fi
```

```
# Sondes
printf "Sauvegarde des sondes \n";
cp -ra /usr/lib/nagios/plugins ${rep}

# Modules Perl
#printf "Sauvegarde des modules PERL \n";
#cp -ra /usr/lib/perl5/ ${rep}

# Modules Python
printf "Sauvegarde des modules Python \n";
cp -ra /usr/lib/python2.6/ ${rep}

# Images
printf "Sauvegarde des images \n";
cp -ra /usr/share/centreon/www/img ${rep}

# Fichier RRD
printf "Sauvegarde des fichiers RRD \n";
cp -ra /var/lib/centreon/status ${rep}
cp -ra /var/lib/centreon/metrics ${rep}

# Fichiers MIBS
printf "Sauvegarde des fichiers de MIBS \n";
cp -ra /usr/share/snmp/mibs ${rep}

# Fichier de conf HTTP
printf "Sauvegarde des fichiers de configuration Apache \n";
cp -ra /etc/httpd/conf ${rep}httpdconf/
cp -ra /etc/httpd/conf.d ${rep}httpdconf/

# Fichier PHP.INI
printf "Sauvegarde du fichier php.ini \n";
cp -ra /etc/php.ini ${rep}

# Clé privée + publique
#printf "Sauvegarde des clés publique et privées de root \n";
#cp -ra /root/.ssh/ ${rep}rootkeys/

# Compression
printf "Compression de la sauvegarde \n";
tar czvf ${dest} ${rep}

# Chongement propriétaire
Chown backup:backup ${dest}

# Suppression des fichiers non compressé
printf "Suppression du repertoire \n";
rm -rf ${rep}*

# Message de fin
printf "Sauvegarde terminée, le fichier de sauvegarde est ${dest} \n";
```

10.2 Sauvegarde des BDD MySQL

Le script `sauv_mysql.sh`.

```
#!/bin/bash

if [ $# -eq 1 ]
then
    DATE=`date +%y%m%d`
    rep=/var/spool/centreon/dump_mysql/
    dest=/home/backup/sauvegardes/dump_mysql/dump_mysql.${DATE}.tar.gz

    # Création du répertoire à compresser
    mkdir ${rep}
    # Suppression de l'ancienne sauvegarde
    rm -f ${dest}

    # On compte le nombre d'archives presentes dans le dossier
    NbArchive=$(ls -A /home/backup/sauvegardes/dump_mysql/ |wc -l)
    # S'il y a plus de 4 archives, on supprime la plus ancienne
    if [ ${NbArchive} -gt 7 ]
    then
        # On recupere l'archive la plus ancienne
        Old_backup=$(ls -lrt /home/backup/sauvegardes/dump_mysql/ |grep ".tar."
→gz" |head -n 1 | cut -d":" -f 2 | cut -d" " -f 2)
        # On supprime l'archive la plus ancienne
        rm -f /home/backup/sauvegardes/dump_mysql/${Old_backup}
    fi

    # On met les DB a sauvegarder dans une variable:
    names=( centreon centreon_storage centreon_status )

    # On bloque les tables le temps de la replication
    # A FAIRE

    for name in ${names[@]}
    do
        printf "Sauvegarde de la BDD : $name \n";
        mysqldump -u backup --single-transaction -p$1 $name > ${rep}$name.sql
    done

    # Compression
    printf "Compression de la sauvegarde \n";
    tar czvf ${dest} ${rep}

    # Changement de proprietaire
    chown backup:backup ${dest}

    # Suppression des fichiers non compressé:
    printf "Suppression du repertoire \n";
    rm -rf ${rep}

    # Message de fin
    printf "Sauvegarde terminée, le fichier de sauvegarde est ${dest} \n";
else
    printf "Utilisation : ./sauv_mysql.sh <mdp_mysql_backup>, \nil est preferable_
→d'executer ce script avec l'utilisateur root";
fi
```

Ensuite, nous planifions une tâche “cron”. Cron est un programme qui permet aux utilisateurs des systèmes Unix d’exécuter automatiquement des scripts, des commandes ou des logiciels à une date et une heure spécifiées à l’avance, ou

selon un cycle défini à l'avance.

Pour vérifier, les tâches en cours il suffit de taper la commande suivante :

```
# crontab -L
```

Pour l'éditer nous devons taper :

```
# crontab -e
```

Cette commande va ouvrir l'éditeur de texte VIM pour éditer les tâches.

Voici les tâches en cours sur le serveur au moment de l'écriture de la documentation :

```
# m h dom mon dow command
#Sauvegarde conf centreon tous les jours à 00h01
01 00 * * * /home/<user>/script_sauvegarde/sauvegarde.sh
# Sauvegarde MySQL tous les jours à 00h30
30 00 * * * /home/<user>/script_sauvegarde/sauv_mysql.sh backup
```

10.3 Récupération des sauvegardes

Disons que nous voulons récupérer cette sauvegarde depuis un poste client Windows (oui, oui ça peut arriver !!)

Pour pouvoir faire cette sauvegarde nous allons tout d'abord créer un compte, sur le serveur qui ne pourra faire que cela.

10.3.1 Coté serveur

Sur le serveur :

```
# useradd backup
# passwd backup
```

On crée l'utilisateur backup et on lui assigne comme mot de passe 'backup'

Ensuite, nous créons un bash restreint ('rbash'), si pas déjà créé.

```
# cp /bin/bash /bin/rbash
```

Voici les restrictions de rbash :

1. Changing directories with cd
2. Setting or unsetting the values of SHELL, PATH, ENV, or BASH_ENV
3. Specifying command names containing /
4. Specifying a file name containing a / as an argument to the . builtin command
5. Specifying a filename containing a slash as an argument to the -p option to the hash builtin command
6. Importing function definitions from the shell environment at startup
7. Parsing the value of SHELL_OPTS from the shell environment at startup
8. Redirecting output using the >, >|, , >&, &>, and >> redirection operators
9. Using the exec builtin command to replace the shell with another command
10. Adding or deleting builtin commands with the -f and -d options to the enable builtin command
11. Using the enable builtin command to enable disabled shell builtins

12. Specifying the -p option to the command builtin command

13. Turning off restricted mode with set +r or set +o restricted

On donne des droits plus élevés à nos fichiers de configuration de l'utilisateur pour qu'il ne puisse pas les modifier :

```
# chown root. /home/backup/.bash_profile
# chmod 755 /home/backup/.bash_profile
# chown root. /home/backup/.bashrc
# chmod 755 /home/backup/.bashrc
```

Puis on change sa variable d'environnement pour qu'il ne puisse utiliser aucune commande :

```
vim /home/backup/.bash_profile
```

On change la ligne :

```
PATH=$PATH:$HOME/bin
```

Par :

```
PATH:$HOME/bin
```

L'utilisateur n'a plus accès aux commandes système, il ne peut rien faire.

10.3.2 Coté client

ur le poste Windows, nous allons créer des dossiers pour accueillir les sauvegardes. Dans le dossier Documents de l'utilisateur nous créons les dossiers suivant : sauvegardes/centreon sauvegardes/mysql

Dans le dossier "centreon", nous récupérerons les sauvegardes des fichiers de configuration, des plugins ... (dump_centreon...tar.gz) Dans le dossier "mysql", les sauvegardes des BDD (dump_mysql...tar.gz)

Le script de sauvegarde est dans le dossier, "Documents/scripts/Script_sauvegarde" Pour ce script, nous utilisons le programme "pscp.exe" qui se trouve sur le "Bureau".

Voici le script sauv_centreon.bat.

```
cd "C:\Documents and Settings\\Documents\sauvegardes\centreon"

"C:\Documents and Settings\\Bureau\pscp.exe" -unsafe -scp -pw backup backup@IP:/
↪home/backup/sauvegarde/dump_centreon/* .

cd "C:\Documents and Settings\\Documents\sauvegardes\mysql"

"C:\Documents and Settings\\Bureau\pscp.exe" -unsafe -scp -pw backup backup@IP:/
↪home/backup/sauvegarde/dump_mysql/* .
```

10.3.3 Tâche planifiée

Maintenant, nous allons créer une tâche planifiée sur le client windows.

Menu démarrer ==> Panneau de configuration ==> Tâches planifiées ==> Création d'une tâche planifiée.

L'assistant pour les tâches planifiées apparaît :

- Cliquer sur "Suivant"
- Parcourir...
- Récupérer le script de sauvegarde sur le Documents/scripts/Script_sauvegarde/sauv_centreon

- Puis on choisi d'exécuter cette tâche tous les jours
- Cliquer sur "Suivant"
- Heure de début = 01 :00
- Cliquer sur "Suivant"
- Entrer le mot de passe de session

La tâche est opérationnelle et sera lancé tous les jours à 1h00 du matin.

CHAPITRE 11

Création utilisateur MySQL

Pour que la sauvegarde des BDD fonctionne, nous devons (pour des raisons de sécurité) créer un utilisateur qui n'aura que des droits pour la sauvegarde. Il suffit de taper les commandes suivantes :

```
$ mysql -u root -p
Enter password : <mot_de_passe_root_MySQL>
```

```
MariaDB [(none)]> CREATE USER 'backup'@'localhost' IDENTIFIED BY 'backup';
MariaDB [(none)]> GRANT SHOW DATABASES, SELECT, LOCK TABLES, RELOAD ON *.* TO 'backup
↪ '@'localhost';
MariaDB [(none)]> FLUSH PRIVILEGES;
MariaDB [(none)]> exit;
```

Sauvegarde configuration routeurs

Pour la sauvegarde de la configuration des routeurs, nous avons 2 possibilités :

- Sauvegarde automatique
- Sauvegarde manuelle

Pour notre projet, une sauvegarde manuelle sera effectué. En effet, la configuration de notre réseau ne nous permet pas d'installer les paquets, modules nécessaire pour une sauvegarde automatique, et de plus cela permettra de travailler avec la commande `expect`. Pour sauvegarder les configurations, nous devrons lancer manuellement le script puis entrer les mots de passe.

Voici le pas à pas :

1. *Création d'un fichier host*
2. *Ajout d'un nouvel hôte*
3. *Ajout des paquets utiles au script*
4. *Création d'un script*
5. *Lancement du script*

12.1 Création d'un fichier host

Pour le script, nous avons besoin d'un fichier contenant tous les routeurs de notre réseau, pour cela, nous allons utiliser un fichier présent dans tous les systèmes d'exploitation GNU/Linux, le fichier `/etc/hosts`

```
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.10.121      TOTO

# Liste Routeurs
192.168.0.1      RT_2001
192.168.0.2      SW_2002
192.168.0.3      RT_2003
192.168.0.7      RT_2007
192.168.0.8      RT_2008
```

```
192.168.0.9      RT_2009
192.168.0.11     RT_2011
192.168.0.12     RT_2012
192.168.0.13     RT_2013
192.168.0.15     RT_2015
192.168.0.16     RT_2016
192.168.0.17     RT_2017
192.168.0.18     RT_2018
192.168.0.21     RT_2021
192.168.0.22     RT_2022
192.168.0.23     RT_2023
192.168.0.24     RT_2024
192.168.0.27     RT_2027
192.168.0.28     RT_2028
192.168.0.33     RT_2033
192.168.0.34     RT_2034
192.168.0.35     RT_2035
192.168.0.36     RT_2036
#192.168.0.37    RT_2037
192.168.0.39     RT_2039
192.168.0.40     RT_2040
192.168.0.43     RT_2043
192.168.0.47     RT_2047
192.168.0.48     RT_2048
```

On peut voir dans ce fichier que nous avons l'adresse IP qui est en relation avec le nom du routeur. Lorsque nous faisons un ping RT_2001 la requête aboutie, (normal en même temps !)

12.2 Ajout d'un nouvel hôte

Lors de l'ajout d'un nouveau routeur/commutateur, il est important de remplir le fichier précédent de la façon suivante : Avec VIM

```
# vim /etc/hosts
```

Pour accéder à la fin du fichier il suffit de taper :

```
Ctrl+G
o (la lettre o en minuscule)
```

Nous entrons en mode INSERTION Puis taper :

```
192.168.0.xxx    RT_20xx
```

Avertissement : L'espace est une tabulation

Puis enregistrer le fichier :

```
Echap ==> :x
```

12.3 Ajout des paquets utiles au script

Pour le bon fonctionnement du script, nous avons eu besoin d'installer les paquets suivant

- tcl
- expect
- telnet

Tcl est une dépendance du paquet expect, c'est un langage de script.

“Expect est un outil d'automation et de tests de non-regression, ..., comme extension au langage de script Tcl pour tester des applications interactives comme telnet, passwd, fsck, rlogin, ssh ou bien d'autres.”¹

C'est-à-dire qu'il va permettre de répondre, d'une façon interactive, aux demandes de login et mot de passe lors de notre connexion en telnet à nos équipements.

Telnet est un protocole qui permet de communiquer entre équipements, attention ce protocole n'est pas du tout sécurisé, il est recommandé d'utiliser SSH à la place, mais bon pour certains projet, on a pas le choix ...

Pour installer ces paquets il suffit de récupérer ceux-ci sur internet dans les dépôts officiels de notre système d'exploitation et de lancer la commande suivante :

```
# rpm -ivh tcl-8.5.7-6.el6.x86_64.rpm
```

Et idem pour les 2 autres.

12.4 Création d'un script

Ce script demande les mots de passe pour se connecter aux routeurs, ainsi que le mot de passe enable pour les équipements CISCO. Il récupère le nom des routeurs dans le fichier `/etc/hosts`, Puis il communique avec l'équipement en telnet et envoie les informations de la configuration dans un fichier au nom de l'équipement dans le dossier `/home/<user>/sauvegardes/routeurs/` Enfin il édite le fichier en enlevant les informations superflues.

Voici le script `sauv_conf_routeurs.sh` : “En commentaire”, les explications.

```
#!/bin/bash
#sauv_conf_routeurs.sh

echo "Veuillez donner le mot de passe pour telnet"
# Permet de cacher la saisie du mot de passe
stty -echo
read pwdTelnet
# Reviens au mode saisie normal
stty echo

echo "Veuillez donner le mot de passe enable"
stty -echo
read pwdEnable
stty echo

export telnet='./telnet.sh'
# On recupere le nom des routeurs
export routeurs=`cat /etc/hosts | sed -e '/^#\|^127\|:::1\|10\.241\.103\.221\|^$/d'|_
↵cut -f2`
export pwdTelnet
export pwdEnable
```

Tiré de la page Wikipédia.

```

export temp='./tmp_routeur.log'

for routeur in $routeurs
do
    rm -f $telnet
    echo $routeur
    export routeur

    case $routeur in
        # On prend les routeurs Juniper (pas meme commande telnet), IL FAUT REMPLIR
        ↪ CETTE LIGNE A CHAQUE AJOUT DE ROUTEUR JUNIPER !!!
        RT_2035|RT_2036|RT_2047|RT_2048)
            # On cree un fichier tenet.sh avec les commandes a envoyer au
            ↪ routeur

            echo 'expect 2>&1 << EOF'>> $telnet
            echo 'spawn telnet $routeur'>> $telnet
            echo 'sleep 5' >> $telnet
            echo 'expect "login:"' >> $telnet
            echo 'send "admin\r"' >> $telnet
            echo 'expect "Password:"' >> $telnet
            echo 'send "${pwdTelnet}\r"' >> $telnet
            echo 'expect ">"' >> $telnet
            echo 'send "show configuration | no-more\r"' >> $telnet
            echo 'expect ">"' >> $telnet
            echo 'send "exit\r"' >> $telnet
            echo 'expect "closed"' >> $telnet
            echo 'exit' >> $telnet
            echo 'EOF' >> $telnet

            ;;

        \*)
            # Ici tous les routeurs CISCO
            echo 'expect 2>&1 << EOF'>> $telnet
            echo 'spawn telnet $routeur'>> $telnet
            echo 'sleep 5' >> $telnet
            echo 'expect "Password:"' >> $telnet
            echo 'send "${pwdTelnet}\r"' >> $telnet
            echo 'expect ">"' >> $telnet
            echo 'sleep 5' >> $telnet
            echo 'send " enable\r"' >> $telnet
            echo 'expect "Password:"' >> $telnet
            echo 'send "${pwdEnable}\r"' >> $telnet
            echo 'expect "#" >> $telnet
            # Pour ne pas avoir le resultat coupe par un "MORE", reviens
            ↪ a la normal apres reconnexion
            echo 'send " terminal length 0\r"' >> $telnet
            echo 'expect "#" >> $telnet
            echo 'send " show running-config\r"' >> $telnet
            echo 'expect "#" >> $telnet
            echo 'send " exit\r"' >> $telnet
            echo 'expect "closed"' >> $telnet
            echo 'exit' >> $telnet
            echo 'EOF' >> $telnet

            ;;

    esac

    # On change les droits sur le fichier pour pouvoir le lancer
    chmod 755 $telnet
    # On lance la commande et on met les résultats dans le fichier $routeur

```

```

$telnet > ../sauvegardes/routeurs/$routeur 2>&1

# On nettoie le fichier
eraseDebut=`grep -n version ../sauvegardes/routeurs/$routeur | cut -d":" -f1`
sed -i "1,${eraseDebut}d" ../sauvegardes/routeurs/$routeur
eraseFIN=`grep -n exit ../sauvegardes/routeurs/$routeur | cut -d":" -f1`
sed -i ${eraseFIN},10000d ../sauvegardes/routeurs/$routeur
done

```

12.5 Lancement du script

Pour lancer le script, il suffit de se connecter sur le serveur avec Putty (si PC Windows), sur le serveur 168.192.10.121
Ensuite se rendre sur /home/<user>/script_sauvegarde

On lance le script de la façon suivante :

```
# ./sauv_conf_routeurs.sh
```

Lorsque l'on lance la commande, celle-ci demande le mot de passe Telnet :

```
Veillez donner le mot de passe pour telnet
```

Puis le mot de passe pour le mode enable :

```
Veillez donner le mot de passe enable
```

Ensuite le script tourne tout seul et récupère les configurations de nos routeurs/commutateurs.

Les fichiers de sauvegarde se trouvent dans le dossier suivant :

```
/home/<user>/sauvegardes/routeurs/
```

Et les fichiers sont au nom de l'équipement, ex : RT_2001

CHAPITRE 13

Installation de NagVis

NagVis va permettre de cartographier notre infrastructure et de voir en temps réel celle-ci sur une carte représentant notre réseau.

Nous allons installer les paquets à la main, le réseau de notre projet n'est pas connecté à Internet. Si votre serveur est connecté au réseau, il est conseillé d'utiliser les dépôts. Par contre, nous avons quand un dépôt local, mais qui ne contient pas les paquets que nous devons installer.

13.1 Installation du broker MK Livestatus

Pour l'installer, nous devons au préalable installer un broker qui fera le lien entre NagVis et Centreon, nous allons donc récupérer MK Livestatus (version 1.2.4p5) au format «tar.gz» que nous placerons sur notre serveur de supervision (10.220.0.16) dans le dossier «usr/local». Nous devons compiler celui-ci et pour se faire, nous aurons besoin de certains paquets :

- gcc
- gcc-c++
- kernel-devel

Pour les installer il faudra taper en ligne de commande :

```
# yum install gcc gcc-c++ kernel-devel
```

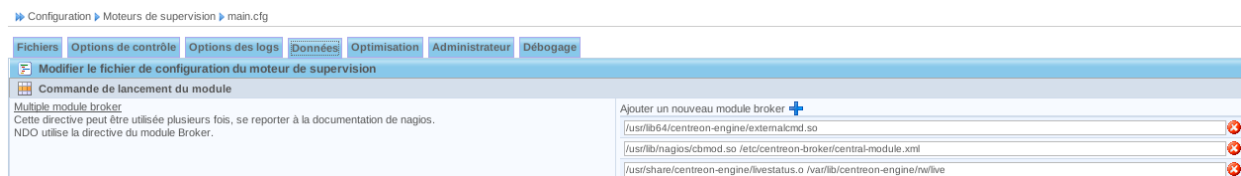
Nous pouvons commencer la compilation, tout d'abord il faut décompresser l'archive, puis nous pourrions compiler :

```
# tar zxvf mk-livestatus-1.2.4p5.tar.gz
# cd mk-livestatus-1.2.4p5
# ./configure
# make
```

Une fois la compilation terminée, vous devrez placer le binaire du broker dans le dossier du moteur de supervision. Vous ajouterez l'utilitaire unixcat permettant la vérification du socket Unix.

```
# cp src/livestatus.o /usr/share/centreon-engine
# cp src/unixcat /usr/share/centreon-engine
```

13.2 Configuration de l'interface Centreon



Onglet Data, rajoutez un module broker en cliquant sur la croix. Saisir la ligne suivante :

```
/usr/share/centreon-engine/livestatus.o /var/lib/centreon-engine/rw/live
```

Ensuite, appliquez la configuration et vérifiez le bon fonctionnement du module en visualisant le fichier de log `/var/log/centreon-engine/centengine.log`

```
[1337663836] livestatus: Livestatus 1.1.12p7 by Mathias Kettner. Socket: '/var/lib/
↪centreon-engine/rw/live'
[1337663836] livestatus: Please visit us at http://mathias-kettner.de/
[1337663836] livestatus: Hint: please try out OMD - the Open Monitoring Distribution
[1337663836] livestatus: Please visit OMD at http://omdistro.org
[1337663836] livestatus: Cannot open log archive '/usr/local/nagios/var/archives/'
[1337663836] livestatus: Finished initialization. Further log messages go to /var/log/
↪centreon-engine/livestatus.log
[1337663836] Event broker module '/usr/share/centreon-engine/livestatus.o' ↪
↪initialized successfully.
[1337663836] livestatus: Timeperiod cache not updated, there are no timeperiods (yet)
```

Par défaut, MK livestatus crée un fichier de log dans `/var/log/centreon-engine/livestatus.log`. Nous pouvons vérifier le fonctionnement du broker avec cette commande. Vous devrez avoir un résultat.

Attention, utilisez l'utilitaire `unixcat` que vous avez compilé mais pas celui fournit avec la distribution Debian. Il ne fonctionne pas avec MK livestatus.

```
echo "GET hosts" | /usr/local/nagios/bin/unixcat /var/lib/centreon-engine/rw/live
```

Puis redémarrer le moteur de supervision.

13.3 Installation de Graphviz

Nous devons récupérer ce paquet pour permettre à NagVis de créer ses cartes. L'archive à récupérer est `graphviz-2.38.0.tar.gz` que nous copions dans `/usr/local`.

```
# cd /usr/local
# tar zxvf graphviz-2.38.0.tar.gz
# cd graphviz-2.38.0
# ./configure
# make
# make install
```

13.4 Installation de NagVis

Continuons par l'installation de Nagvis. Nous utiliserons la dernière version 1.7.10 pour la version stable. Récupérez l'archive (`nagvis-1.7.10.tar.gz`) pour la copier dans `/usr/local`

```
# cd /usr/local
# tar zxvf nagvis-1.7.10.tar.gz
# mv nagvis-1.7.10 nagvis
# cd nagvis
```

Lorsque nous sommes dans ce fichier, nous devons lancer la commande d'installation.

```
# ./install.sh -l unix:/var/lib/centreon-engine/rw/live
```

L'option `-l` permet de forcer l'installation du socket unix hors du dossier par défaut.

Voici les réponses à donner lors des quelques questions :

1. `proceed ==> y`
2. `nagios base directory ==> /usr/share/centreon-engine`
3. `path to NagVis ==> /usr/local/nagvis`
4. `Do you want to use backend mklivestatus ? ==> y (et n pour les autres)`
5. Répondre oui pour la suite ainsi que les réponses par défaut.

Si problème avec `mklivestatus`, on peut changer les informations après l'installation en allant dans `/usr/local/nagvis/etc/nagios.ini.php` et on recherche `backend_live_1` et entrer :

```
socket= «unix:/var/lib/centreon-engine/rw/live»
```

Pour prendre en compte les modifications ainsi que les nouvelles installations, nous devons redémarrer apache :

```
# service httpd restart
```


CHAPITRE 14

Ce qu'il reste à faire

Il reste encore certains aspects dans lesquels j'aimerais rentrer. Je le ferais au fur et à mesure de mon inspiration.

- Les traps
- Les events-handler
- Les pollers
- [La haute-disponibilité](#)
- La réplication MySQL
- La création d'une commande
- Les fichiers de logs utiles
- Et beaucoup d'autres choses

La plupart de ces doc sont déjà écrite, je n'ai plus qu'à les recopier et les intégrer dans cette documentation, qui sera j'espère utile pour certains.

Vous pouvez également aller voir mon pense bête sur l'utilisation de [Git](#)

CHAPITRE 15

Indices and tables

- `genindex`
- `modindex`
- `search`