TP n°12 – Gestion des services

1) Configuration et utilisation du service rsyslog

Le service rsyslog permet de gérer tous les messages d'information et d'erreurs du système. Lire ou relire le cours pour les détails. Les messages de log sont enregistrés dans différents fichiers du dossier /var/log : messages, user.log, auth.log...

Les messages sont produits par des programmes C, ou par la commande shell `logger`. On doit indiquer le « facility » et le « severity », par exemple user.info. Voici un exemple, tapez ceci :

```
logger -p user.info "Salutations"
```

Puis allez regarder la fin de ces fichiers :

```
sudo ls /var/log
sudo tail /var/log/messages
sudo tail /var/log/user.log
sudo tail /var/log/syslog
```

Pourquoi tous ces fichiers ? Pour bien ranger les messages selon les catégories. NB : ces fichiers sont protégés de la lecture par n'importe qui car il se peut qu'il y ait des informations sensibles. Notez qu'il y a la date, l'heure et toutes les informations permettant d'identifier la source du message.

Cela est configuré par le fichier `/etc/rsyslog.conf`. Il faut se rendre à la partie #### RULES ####. Ses lignes sont composées de deux parties : à gauche, il y a un filtre : *facility.level* (l'un et l'autre pouvant être jokérisé), et à droite il y a un nom de fichier, celui qui reçoit le message. S'il y a un – devant le nom de fichier, ça indique qu'il ne faut pas forcer une écriture disque immédiatement (par contre, si le système plante, alors le message est perdu). Noter qu'un message d'un niveau de sévérité donné (ex : err) est envoyé également sur les fichiers de niveaux moins graves (ex : info), sauf s'il y a un = devant la sévérité.

Le même message peut être envoyé dans plusieurs fichiers. D'après la configuration existant sur votre machine, faites la liste des fichiers qui sont remplis si on envoie un message sur « daemon.warn ». Faites un essai (sudo a tous les droits).

```
logger -p daemon.warn "faites attention a ce message"
```

Puis cherchez quels fichiers ont reçu ce message (grep ou tail).

Maintenant, avec vi, rajoutez une ligne dans `/etc/rsyslog.conf` :

```
local0.*                /var/log/local0.log
```

Ensuite, demandez à rsyslog de relire son fichier de configuration :

```
sudo service rsyslog restart
```

Envoyez ensuite un message sur `local0.info` disant que vous faites un essai. Vous devriez le voir dans `/var/log/local0.log`

2) Création et gestion des services

On va étudier les services au travers de la création d'un nouveau service (inutile mais pédagogique). Le TP 13 donnera l'occasion de faire un service un peu plus utile.

a) Démon : /usr/local/bin/demon.sh

Écrire un script tout simple appelé `demon.sh` dans `/usr/local/bin` qui tourne en boucle, sans trop charger la machine et qui calcule régulièrement le nombre de processus de votre machine. Ce

nombre sera envoyé automatiquement sur un fichier de log.

```
#!/bin/bash
while true
do
    nbre=$(ps -edf | wc -l)
    logger -p local0.info "Il y a $nbre processus"
    sleep 30s
done
```

D'abord vérifiez qu'il marche tout seul : lancez-le en arrière-plan, laissez-le tourner une ou deux minutes puis tuez-le. Allez voir les résultats dans `/var/log/local0.log`.

Dans la suite de ce TP, on va en faire un service : créer un lanceur et l'installer dans le système.

b) Lanceur de démon : `/etc/init.d/demon`

Copiez le fichier `/etc/init.d/skeleton` en `/etc/init.d/demon` puis éditez-le, voici les lignes à changer. Si vous éditez avec nano, rajoutez l'option -c pour voir les numéros des lignes.

ligne 3 : mettez `demon` au lieu de `skeleton`

ligne 8 et 9 : mettez « mon démon à moi » en guise de description.

ligne 13 : mettez votre nom à la place de Foo Bar <foobar@baz.org>

ligne 22 : mettez « mon démon à moi » en guise de description.

ligne 23 : mettez `demon` au lieu de `daemonexecutablename`

ligne 24 : la ligne doit devenir `DAEMON=/usr/local/bin/demon.sh` (le nom complet du démon)

ligne 25 : videz la chaîne entre " "

Comprendre comment marche ce lanceur : son paramètre $1 vaut soit « start », soit « stop ». Si c'est « start », alors ça va dans la fonction `do_start` qui lance le démon en arrière-plan et note son PID dans le fichier `/var/run/demon.pid`. Si c'est « stop », alors ça va dans la fonction `do_stop` qui tue le processus dont le PID est dans `/var/run/demon.pid` puis supprime ce fichier.

Dans ces deux fonctions, `do_start` et `do_stop`, c'est la commande `start-stop-daemon` qui travaille. Regardez ses paramètres : `--start` ou `--stop`. Les autres paramètres indiquent comment on le lance le démon : en arrière-plan, en créant un fichier de PID et en lançant l'exécutable dont le nom est dans la variable DAEMON. Pour l'arrêt dans `do_stop`, c'est également très parlant : on essaye de tuer le démon avec son fichier de PID.

Rendez ce lanceur exécutable : `chmod a+x /etc/init.d/demon`

c) Déclaration du service pour systemd

Il reste un fichier à créer, pour le mécanisme systemd qui gère maintenant les services Unix.

Créez le fichier `/etc/systemd/system/demon.service` (en mode sudo) et mettez ceci dedans :

```
[Unit]
Description=demon pour tester les services

[Service]
ExecStart=/usr/local/bin/demon.sh

[Install]
WantedBy=multi-user.target
```

Testez en faisant :

```
sudo service demon start
ps -edf
sudo service demon status
sudo service demon stop
ps -edf
sudo service demon status
```

Que se passe-t-il si on lance deux fois le démon... testez et regardez avec ps -edf.

Que se passe-t-il si on essaye d'arrêter deux fois de suite le démon ?

Ce qu'il faudrait essayer, c'est de tuer le processus du démon par un kill -9 et voir ce que `sudo service demon status` affiche.

REMARQUE 1 : avec systemd, maintenant, le lanceur `/etc/init.d/demon` n'est plus utilisé pour un lancement manuel. Il ne sert que pour le démarrer automatiquement, voir le § suivant.

REMARQUE 2 : dans les nouvelles versions de Debian, on doit utiliser la commande `systemctl` pour piloter les services :

```
sudo systemctl start demon
sudo systemctl status demon
sudo systemctl stop demon
```

J'ai laissé les anciennes commandes parce qu'elles s'appliquent à de nombreux autres systèmes Unix qui ne sont pas encore passés à systemd.

d) Lancement automatique

On va maintenant demander au système de lancer le démon automatiquement au démarrage de la machine et de l'arrêter à l'extinction. Pour cela, le système se sert d'un lien appelé S*NN*demon dans /etc/rc5.d et des liens appelés K*MM*demon dans /etc/rc0.d et /etc/rc6.d ; *NN* et *MM* sont des « rangs » qui donnent l'ordre de lancement ou d'arrêt. Ces liens sont gérés par ce qui s'appelle les runlevels, sachant que 0 est celui de l'extinction, 6 celui du redémarrage et 5 celui du fonctionnement normal.

La commande `runlevel` (en mode sudo) affiche le runlevel actuel.

Il faut faire comme ça pour créer les liens :

```
sudo update-rc.d demon defaults
ls -l /etc/rc*.d/*demon
ls -l /etc/rc5.d
ls -l /etc/rc0.d
```

defaults indique d'utiliser les informations LSB du lanceur : les lignes `Default-Start` et `Default-Stop` au début du script.

On voit que dans le dossier /etc/rc5.d, le système (INIT) va démarrer successivement rsyslog et d'autres services de rang 1 (*NN*=01), puis ensuite votre démon en rang 2 ou 3 (*NN*=02). C'est dû à la ligne `Required-Start` qui est au début du lanceur : remote_fs et syslog. Et dans le dossier /etc/rc0.d, votre démon sera tué (Kill) parmi les premiers.

Pensez à sauver votre Wiki maintenant : vérifiez qu'il y a bien eu « changes saved ».

Il faut maintenant tester, faites sudo reboot et attendez le redémarrage complet puis vous vérifierez qu'il y a bien, maintenant, un processus démon (ps et coup d'œil au logs)

Vous pouvez le stopper à tout moment avec

```
sudo /etc/init.d/demon stop
```

Et vous pouvez supprimer le lancement automatique avec cette commande qui supprime les liens :

```
sudo update-rc.d -f demon remove
```

Si vous laissez ce démon en service, vous le retrouverez dans vos pattes au prochain TP, sans doute ça vous agacera un peu.

3) Configuration du service cron

Le démon *cron* gère un emploi du temps et déclenche des commandes aux moments prévus. L'emploi du temps s'appelle un *crontab*. Chaque utilisateur peut posséder son propre *crontab*. Les *crontabs* sont stockés dans `/var/spool/cron`. Bien que ce soient des fichiers texte, on ne peut pas les éditer n'importe comment, il faut faire appel à la commande `crontab` :

`crontab -l` liste les éléments du *crontab* (vide pour lili initialement)

`crontab -e` appelle l'éditeur vi pour modifier le crontab

On va d'abord vérifier que fcron fonctionne : tapez `crontab -e` (ça lance vi) et mettez la ligne suivante, attention, d'abord taper la commande `tty` pour savoir quel est votre terminal : /dev/tty1 ou /dev/pts/0, vous devez mettre ce que vous affiche la commande `tty`, sinon rien ne sera visible.

```
*  *  *  *  *  date > /dev/tty1 ou /dev/pts/0
```

NB1 : il y a au moins un espace entre les 5 *, en fait partout où en voit un.

NB2 : vérifiez que votre terminal est bien `/dev/tty1` (travail en mode texte dans la console) ou `/dev/pts/0` (travail en mode X11) à l'aide de la commande `tty`.

Cette directive demande d'afficher la date et l'heure toutes les minutes sur votre terminal : attendez que ça arrive. Ensuite, une fois que vous en aurez assez, vous pouvez enlever cette directive en relançant `crontab -e` et en supprimant la ligne.

Vous comprenez qu'on ne va pas pouvoir tester les 5 champs : minutes, heures... on n'a pas le temps d'attendre pour vérifier que ça marche.

Éventuellement, vous pourriez utiliser ce mécanisme pour améliorer le démon vu dans le § précédent. On n'a pas le temps malheureusement de tout essayer.

4) Configuration du service logrotate

C'est un autre service du système. Il sert à vider ou compacter les fichiers de logs tous les jours afin de ne pas saturer les disques de vieilleries.

On va configurer la rotation du fichier `local0.log`. Il suffit de créer un fichier `/etc/logrotate.d/local0` contenant :

```
/var/log/local0.log {
    daily
    rotate 4
    missingok
    notifempty
}
```

Il dit : faire tourner chaque jour, garder les 4 dernières versions, et c'est pas grave s'il est absent, et ne rien faire s'il est vide.

On peut forcer son exécution par ces commandes (à répéter plusieurs fois) :

```
logger -p local0.info "essai de logrotate"
ls /var/log/local0.*
sudo logrotate -f /etc/logrotate.conf
ls /var/log/local0.*
```

Constatez que le fichier local0.log a fait l'objet d'un roulement, seulement s'il y a eu un changement dedans.

Mais en général, il est déclenché par un crontab par exemple chaque jour à midi dix :

```
10 12 * * *  /usr/sbin/logrotate /etc/logrotate.conf
```

Par contre, vous comprenez qu'on ne peut pas le tester dans un TP de 2h.

C'était copieux... en fait, l'administration système, c'est toujours un peu comme ça : des tas de commandes, d'options et de fichiers de configuration, mais quand ça marche on est content.