Principes et pratiques de l'administration des systèmes

Module 3I015

Semaine 5

1. Sauvegardes des comptes utilisateurs avec `rsync`

`rsync` (ou *remote synchronization* en anglais) est un logiciel très simple à utiliser et à configurer. Il va vous permettre d'effectuer des sauvegardes différentielles : on ne met à jour que ce qui a changé depuis la dernière sauvegarde. Ainsi, la première sauvegarde des comptes utilisateurs peut être un peu longue mais les suivantes sont très rapides, du fait que l'on ne prend en compte que les différences avec la dernière sauvegarde réalisée (nouveaux fichiers, fichiers détruits, fichiers modifiés).

Plusieurs alternatives s'offrent quant à l'utilisation de `rsync` :

1. soit msLDAP envoie les données à sauvegarder à ssLDAP via le réseau que vous avez créé (par connexion sécurisée SSH),
2. soit ssLDAP sauvegarde lui-même les comptes utilisateurs en passant par le réseau via une connexion sécurisée SSH,
3. soit ssLDAP procède lui-même en local à la sauvegarde puisque `/users` est monté sur ssLDAP.

Dans ce TME, pour simplifier, nous utiliserons la troisième option. L'utilisateur `root` de ssLDAP lancera donc la commande `rsync`, qui sauvegardera directement ses données dans le répertoire `/backup`. Ce répertoire contiendra donc une copie, un miroir, de `/users`.

Comme nous souhaitons avoir à disposition plusieurs sauvegardes, nous allons compresser les fichiers transmis par `rsync`. Pour cela, nous avons, là encore, plusieurs options : la première consiste à utiliser une archive compressée de type `tar.gz` pour stocker l'ensemble des fichiers des utilisateurs. Ce n'est pas forcément la meilleure option car le fait de n'avoir qu'un seul « gros » fichier à transférer, qui sera modifié d'une sauvegarde à l'autre, implique que `rsync` devra transmettre de grosses quantités de données à chaque sauvegarde (tout le fichier `tar.gz`), ce qui anéantit *de facto* l'intérêt des sauvegardes différentielles. La deuxième option consiste à copier les répertoires des utilisateurs tels quels mais sur un système de fichiers compressé. Vous avez actuellement créé une partition `/backup` que vous avez formatée en `ext4`, et elle n'est pas compressée. Nous vous proposons dans la suite de la reformater en `btrfs`, un autre système de fichiers, et de la monter avec son option de compression. Celle-ci est transparente pour l'utilisateur : vous pourrez accéder aux fichiers de `/backup` de la même manière qu'avec vos autres partitions `ext4`, votre système Linux compressera et décompressera à la volée les fichiers de `/backup` que vous manipulerez. Évidemment, ces opérations augmentent les temps d'accès aux fichiers mais ce n'est pas très pénalisant ici car les sauvegardes des comptes des utilisateurs ont en principe lieu la nuit, quand les utilisateurs et l'administrateur système dorment, et les restaurations de fichiers sont peu fréquentes.

Étape 1 – Reformatage de `/backup` sur ssLDAP

Afin de reformater la partition qui contient `/backup`, il convient tout d'abord de la démonter. Comme celle-ci est exportée vers msLDAP, il faut d'abord arrêter, temporairement, votre serveur NFS sur ssLDAP, puis réaliser le `umount /backup`.

Pour formater votre partition en `btrfs`, il vous faut installer via `apt-get` le package « `btrfs-tools` ». Ensuite, il vous suffira de réaliser un `mkfs -t btrfs -f /dev/sda`X pour effectuer le formatage (X = le numéro de la partition sur laquelle se trouve `/backup`). Dans votre `/etc/fstab`, remplacez `ext4` par `btrfs` pour le type de votre partition et rajoutez l'option `compress` pour activer sa compression.

Remontez votre `/backup` et redémarrez votre serveur NFS.

Étape 2 – Vérification de la compression

Sur ssLDAP, tapez la commande `df` afin de visualiser les taux de remplissage des différents « *devices* ». Notez le nombre de kilo octets utilisés dans `/dev/sda1` (votre système de fichiers `/`). La commande `dd` permet de copier des séquences d'octets d'un fichier vers un autre. Utilisez la commande :

```
dd if=/dev/zero of=/root/toto bs=10M count=10
```

afin de créer un fichier `toto` de 100Mo (rempli de « 0 »). Refaites un `df` et vérifiez qu'effectivement `/dev/sda1` utilise 100Mo de plus que précédemment. Notez maintenant le nombre de kilo octets utilisés dans la partition de votre `/backup`. Puis faites un `cp /root/toto /backup`. Attendez quelques secondes et refaites un dernier `df` et observez si le nombre de kilo octets utilisés dans `/backup` est bien nettement inférieur à 100Mo (cela doit être le cas si vous avez activé l'option de compression).

Étape 3 – Installation de rsync

Sur ssLDAP, installez le package `rsync`.

Étape 4 – Sauvegarde différentielle avec rsync

Sur ssLDAP, réalisez toutes les minutes des sauvegardes différentielles de `/users` en utilisant la commande `rsync`. Vous pourrez avantageusement exploiter `cron` pour planifier les sauvegardes toutes les minutes. Vous ferez attention à ce que les propriétés des fichiers et répertoires (propriétaire, groupe, date de dernière modification, droits d'accès, *etc.*) soient préservées.

Préparation avant le TME: *Étudiez le manuel de la commande* `rsync` *afin de déterminer les options dont vous aurez besoin.*

Étape 5 – Tests du bon fonctionnement de rsync

Vérifiez que `root` sur msLDAP a bien accès aux fichiers de `/backup`. Ajoutez un nouveau fichier `toto` dans le home directory de `student1`. Faites une sauvegarde par `rsync` et vérifiez que le nouveau fichier apparaît bien dans votre répertoire de sauvegarde. Supprimez le fichier `toto` sur c1LDAP, refaites une sauvegarde et vérifiez que ce fichier a bien disparu de votre répertoire de sauvegarde.

2. Création d'un nouveau service

L'objectif de cette partie est de créer un service minimaliste afin d'observer les liens entre ce que vous avez déjà fait en programmation système et l'administration de systèmes. Pour cela, vous allez compiler et exécuter un petit serveur, qui se contentera de transmettre la chaîne de caractères « `Message de votre serveur : hello` » aux clients qui se connecteront à lui. Ensuite, vous mettrez en place le mécanisme permettant de démarrer ce serveur automatiquement en tant que service 3i015.

Le code du serveur est fourni dans le fichier `hello.c`, disponible dans le répertoire `/images/ressources` de la machine n°2 de votre salle de TME. Il est également disponible dans la rubrique « Ressources » du site web de l'UE. Si vous souhaitez examiner son code, voici quelques rappels succincts sur les sockets et les architectures client / serveur. Votre serveur et ses clients vont communiquer via ce que l'on appelle des *sockets*, qui sont des canaux de communication inter processus.

Les étapes permettant au **serveur** de proposer ses services sont les suivantes :

1. créer une socket via l'appel système `socket ()`,
2. affecter la socket à une adresse IP et un numéro de port via l'appel système `bind ()`,
3. écouter les connexions des clients via la fonction `listen ()`,
4. accepter les connexions des clients via la fonction `accept ()`,
5. transmettre et recevoir des messages via la socket reliée au client.

Préparation avant le TME: *Vous pouvez lire les documentations des deux URLs suivantes, qui expliquent en détails comment créer un serveur en C :* `http://broux.developpez.com/articles/c/sockets/` *et* `http://www.commentcamarche.net/contents/1053-les-fonctions-de-l-api-socket`.

Étape 6 – Compilation de votre serveur sur msLDAP

Sur msLDAP, téléchargez dans le répertoire `/root` le fichier `hello.c`, disponible dans le répertoire `/images/ressources` de la machine n°2 de votre salle de TME. Compilez-le avec `gcc`.

Étape 7 – Test du bon fonctionnement du serveur

Exécutez le programme en background (cf. le cours sur les jobs). Tapez la commande `netstat -tan`. Vous devriez observer un affichage similaire à :

```
[li350@localhost ~]$ netstat -tan
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale          Adresse distante        Etat
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:2000            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:34161           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:39890           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:38038           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:52696           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:46075           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:2049            0.0.0.0:*               LISTEN
```

```
tcp        0      0 0.0.0.0:389             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:45514           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:40473         127.0.1.1:389           ESTABLISHED
tcp        0      0 127.0.0.1:40471         127.0.1.1:389           ESTABLISHED
tcp        0      0 127.0.1.1:389           127.0.0.1:40468         ESTABLISHED
tcp        0      0 127.0.1.1:389           127.0.0.1:40470         ESTABLISHED
tcp        0      0 127.0.0.1:40468         127.0.1.1:389           ESTABLISHED
tcp        0      0 127.0.0.1:40470         127.0.1.1:389           ESTABLISHED
tcp        0      0 127.0.1.1:389           127.0.0.1:40473         ESTABLISHED
tcp        0      0 127.0.1.1:389           127.0.0.1:40471         ESTABLISHED
tcp6       0      0 :::111                  :::*                    LISTEN
tcp6       0      0 :::40049                :::*                    LISTEN
tcp6       0      0 :::34909                :::*                    LISTEN
tcp6       0      0 :::43101                :::*                    LISTEN
tcp6       0      0 :::45566                :::*                    LISTEN
tcp6       0      0 :::2049                 :::*                    LISTEN
tcp6       0      0 :::59330                :::*                    LISTEN
tcp6       0      0 :::44419                :::*                    LISTEN
tcp6       0      0 :::389                  :::*                    LISTEN
```

La commande `netstat` permet d'afficher les connexions réseau ainsi que les serveurs en écoute (listen) des clients. En particulier, observez que, sur la 2ème ligne débutant par tcp ci-dessus, il y a un serveur sur le numéro de port 2000 (`0.0.0.0:2000`). C'est bien entendu votre serveur.

Vous pouvez vous y connecter en utilisant la commande `telnet 127.0.0.1 2000`, qui demande à telnet de vous connecter sur le serveur de la machine 127.0.0.1 écoutant le port 2000. Vous devriez alors voir apparaître la phrase émise par votre serveur. Pour sortir de telnet, taper sur les touches `ctrl` et `]`, puis taper "quit".

Étape 8 – Arrêt du serveur

Arrêtez votre serveur : utilisez la commande `jobs` pour savoir le numéro de *job* de `hello` et placez-le en *foreground*. Ensuite, arrêtez-le avec un ctrl-c.

Étape 9 – Démarrage automatique de votre serveur : service 3i015

Il s'agit maintenant de faire en sorte que votre serveur soit reconnu comme un service à part entière de votre machine msLDAP. Pour cela, vous pouvez observer que les définitions des services usuels se trouvent dans `/lib/systemd/system` : ce sont tous les fichiers dont l'extension est `.service`. Regardez le contenu du fichier `cron.service`, il est typique :

```
3i015@msLDAP:/lib/systemd/system$ more cron.service
[Unit]
Description=Regular background program processing daemon
Documentation=man:cron(8)

[Service]
EnvironmentFile=-/etc/default/cron
ExecStart=/usr/sbin/cron -f $EXTRA_OPTS
IgnoreSIGPIPE=false
```

```
KillMode=process

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

Ce fichier est divisé en 3 sections : `[Unit]`, `[Service]` et `[Install]` :

- La section `[Unit]` décrit le service, indique où se trouve sa documentation et précise après quels autres services il peut être démarré.
- La section `[Service]` indique comment exécuter le service.
- La section `[Install]` indique sous quelles conditions il faut démarrer le service : la règle `WantedBy=multi-user.target` précise ici que le service doit être démarré si l'on veut atteindre la cible « multi-user ». Si l'on examine attentivement votre système, celui-ci démarre en mode graphique (cible `graphical.target`), qui demande à ce que la cible `multi-user.target` soit atteinte. Par conséquent, la section `[Install]` ci-dessus requiert le démarrage du service `cron` au boot de la machine.

Écrivez un fichier `/etc/systemd/system/3i015.service` qui transformera votre serveur `hello` en service : pour la section `[Unit]`, vous indiquerez qu'il s'agit d'un service de test 3i015 et qu'il doit être démarré après que le réseau ait lui-même été démarré :

```
[Unit]
Description=service de test 3i015
Documentation=http://3i015
After=network-online.target
```

Pour la section `[Install]`, vous ferez en sorte que la cible « multi-user » nécessite le démarrage de votre service. Enfin, pour la section `[Service]`, vous pourrez indiquer :

```
[Service]
Type=simple
ExecStart=/root/hello
```

Étape 10 – Test de votre service 3i015

Exécutez un `systemctl status 3i015`. Vous devriez voir apparaître un message similaire à :

```
3i015.service - service de test 3i015
   Loaded: loaded (/lib/systemd/system/3i015.service; disabled)
   Active: inactive (dead)
     Docs: http://3i015
```

Démarrez votre service 3i015 via la commande `systemctl start 3i015`. En tapant `systemctl status 3i015`, vous pourrez constater que votre service a bien été démarré :

```
3i015.service - service de test 3i015
   Loaded: loaded (/lib/systemd/system/3i015.service; disabled)
   Active: active (running) since jeu. 2015-10-01 17:56:30 CEST; 4s ago
     Docs: http://3i015
 Main PID: 1181 (hello)
   CGroup: /system.slice/3i015.service
           |-1181 /root/hello
```

Vous pouvez également exécuter un `netstat -tan` afin de vous assurer que votre serveur `hello` fonctionne bien.

Exécutez maintenant un `systemctl stop 3i015` afin d'arrêter votre service. Vérifiez via `systemctl status 3i015` et `netstat -tan` que l'arrêt est effectif.

Étape 11 – Pérennisation de votre service 3i015

Exécutez un `systemctl enable 3i015` afin pérenniser le démarrage de votre service. Redémarrez msLDAP puis vérifiez via `systemctl status 3i015` et `netstat -tan` que votre service est bien démarré automatiquement.