

TP 3A - GNU/Linux

Administration des services

Premiers scripts en Bash

Nicolas Delanoue

Exercice 1 (Connexion à un serveur distant via ssh)

Le serveur `wargame.newbiecontest.org` nous offre un accès `ssh` limité à sa machine via le port 22222 avec comme login, mot de passe : `level100`.

1. Trouvez les options et la syntaxe de la commande `ssh` vous permettant de préciser le port et le login.
2. Connectez-vous à cette machine distante.
3. Pouvez-vous lister les fichiers présents dans `/home` ? Que pouvez-vous en déduire ?
4. Quelle est la quantité de mémoire disponible sur cette machine ?
5. De combien de coeurs disposent cette machine ?

Exercice 2 (Un serveur ssh chez soi)

1. Dans un terminal, listez les services qui tournent.
2. Faites un ping vers l'adresse IP `127.0.0.1`. A quoi correspond cette adresse ?
3. Tentez un `ssh 127.0.0.1`, pour quelle raison la connexion n'a-t-elle pas pu aboutir ?
4. Installez et lancez un serveur `ssh`. *Indication* : On pourra faire une recherche comme `apt search sshd` pour connaître le nom du paquet distribuant un démon `ssh`.
5. Listez les services qui tournent.
6. Quelle commande vous permet de vérifier que le service `ssh` est actif ?
7. Donnez les détails du service `ssh`.
8. Reprenez la question 3, la connexion a-t-elle pu aboutir ?
9. Arrêtez-le et faite le test depuis un poste distant pour bien vérifier qu'il est arrêté.

Exercice 3 (Apache, un serveur web)

1. Installez le démon `http` nommé `apache2`.
2. Lancez-le puis vérifiez son fonctionnement avec la commande `systemctl`.
3. Depuis un client firefox en tapant l'adresse `http://127.0.0.1` dans la barre d'url.
4. Visualisez et renseignez-vous sur les différents fichiers de configuration d'Apache 2 (il y en a beaucoup ...).
5. Quel est le port d'écoute d'Apache, et plus généralement du protocole HTTP ? Où trouve-t-on cette information dans les fichiers de configuration d'Apache ?
6. Où trouve-t-on les fichiers de logs par défaut ? Gardez un oeil sur ces fichiers pendant le reste de ce TP.

7. Les fichiers diffusés par apache aux clients sont par défaut dans le répertoire `/var/www/html/`. Créez un fichier `ma_page.html` dans ce répertoire avec comme contenu

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Ma premiere page sous GNU/Linux Apache</title>
5   </head>
6   <body>
7     <p>Trop cool, cela fonctionne !</p>
8   </body>
9 </html>
```

8. Est-il difficile de partager des données via ce canal ? Est-ce sécurisé ?

Exercice 4 (Plannifier une tâche périodique avec cron)

1. Créez ou récupérez en ligne le fichier `/tmp/mon_prog.sh` contenant :

```
1 #!/bin/bash
2 echo Hello I am running $0 at $(date) >> /tmp/historique
```

2. Rendez-le exécutable et testez ce programme.
3. Renseignez-vous sur le service `cron`.
4. Vérifiez que ce service est fonctionnel avec une commande basé sur `systemctl`.
5. Regardez le contenu du fichier `/etc/crontab`. Dans ce fichier, les lignes commençant par `#` sont des commentaires.
6. Créez un fichier `mon_crontab` et ajoutez un ligne dans ce fichier texte de sorte à configurer une tâche nommée `mon_prog.sh` à s'exécuter toutes les 2 minutes. *Indication* : La ligne suivante indique que le programme `/home/user/verifie_disque` devra s'exécuter toutes les 10 minutes avec le compte `user`.
`*/10 * * * * user /home/user/verifie_disque`
7. Ajoutez cette planification au service `cron` avec la commande suivante
`$ crontab mon_crontab`
8. Vérifiez son chargement avec la commande `crontab -l`. Est ce que le contenu du fichier `/etc/crontab` à changer ?
9. Vérifiez que le fichier `/tmp/historique` est bien mis à jour toutes les deux minutes.
10. Quelle option de la commande `crontab` permet de retirer cette planification.

Exercice 5 (Script bash)

Ecrire un script qui :

1. invite l'utilisateur à saisir un nom de répertoire, puis le crée avec la commande `mkdir`,
2. modifie les droits du nouveau répertoire de sorte que tout le monde puisse lire le contenu,
3. en utilisant `touch`, crée plusieurs fichiers vides nommés `fichier1`, `fichier2`, ... `fichier40` et exécute `ls` dessus pour vérifier qu'ils sont présents,
4. met la chaîne caractère "vraiment passionnant ce fichier" dedans en utilisant `echo` et une redirection,
5. affiche leur contenu à l'aide de `cat`,
6. dit au revoir à l'utilisateur.

Exercice 6 (Second script bash)

1. Créez ou récupérez en ligne le fichier `users.txt` dont le contenu est le suivant :

```
1 nicolas
2 julie
3 timothee
4 joris
```

2. Créez un fichier `mon_script.sh` avec comme contenu :

```
1 #!/bin/bash
2 cat < users.txt | while read ligne ; do
3     echo $ligne
4 done
```

3. Exécutez-le et comprenez chacune de ces lignes.

4. En vous appuyant sur le script précédent, développez un script qui :

— télécharge automatiquement la liste des utilisateurs disponible via :

```
http://perso-laris.univ-angers.fr/~delanoue/polytech/3A/gnu_linux/
users.txt
```

— crée, pour chaque utilisateur présent dans ce fichier, un dossier dans le répertoire `/tmp/`

— fabrique, dans chaque répertoire, un fichier nommé `index.html` et basé sur le modèle suivant en remplaçant `user` par le nom de l'utilisateur adéquate.

```
1 <!DOCTYPE html>
2 <html>
3     <body>
4         <p>Bienvenue sur la page de user !</p>
5     </body>
6 </html>
```