Archiver et compresser transférer

1 Archiver les données

Dans cette section, vous allez apprendre à archiver et compresser des fichiers ou dossiers, à transférer des fichiers depuis votre machine vers une machine distante ou inversement et à vous connecter à une machine distante pour effectuer un travail à distance.

1 La commande tar

La commande **tar** (**T**ape **AR**chiving) permet de transformer une hiérarchie de fichiers et/ou de dossiers en une archive prenant la forme d'un seul fichier au format tar. Elle permet aussi d'extraire cette archive. Il faut noter que la commande tar ne permet pas de compresser les données (réduire la taille).

Les utilisations principales de la commande tar sont :

- la sauvegarde. En effet, il plus facile de sauvegarder un seul fichier plutôt qu'une hiérarchie de répertoires. De plus, il est parfois complexe de stocker la hiérarchie de fichiers sur un système de fichiers ne supportant pas tous les noms et attributs de fichiers connus par Unix. Par exemple une clef USB formatée avec le système FAT n'est pas sensible à la casse. Transformer une hiérarchie en un fichier permet de s'affranchir de ces problèmes
- Le transfert. Transférer une hiérarchie de fichiers en un bloc plutôt qu'en transférant de multiples fichiers permet de s'assurer plus facilement d'un transfert correct. Vérifier l'intégrité d'un fichier est beaucoup plus facile que de vérifier l'intégrité de toute une hiérarchie de fichier ou de sous-répertoires.

Les options les plus courantes de la commande tar sont :

–**c** : **c**rée un archive.

–**x** : extraction d' une archive.

–**v** : mode verbeux, affiche la progression.

–**f** : permet de spécifier le nom du fichier d'archive.

L'option **v** indique que l'on veut le mode verbeux qui affiche toutes les opérations effectuées par la commande et liste la hiérarchie de fichiers et sous-répertoires parcourus. L'option **f indique que ce qui suit est le nom de l'archive**.

Il faut donc faire attention car la commande ci-dessous archivera le répertoire X dans une archive nommé v.

```
tar -cfv X
```

- Création d'une archive

```
tar -cvf nom_a_creer.tar chemin_repertoire_existant
```

Le chemin vers le répertoire à archiver ne doit ni commencer par / ni par . ou .. afin que l'on puisse choisir la destination lors de l'extraction.

- Extraction d'une archive :

La commande `tar` est aussi capable d'**extraire** une archive. Option -x

```
tar -xvf archive.tar
```

L'extraction est faite dans le répertoire courant.

Pour extraire le contenu de l'archive dans un **répertoire spécifique**, il faudra ajouter l'option **-C**. Par exemple, pour extraire le contenu du fichier archive.tar dans le répertoire /tmp, entrez la commande suivante :

```
tar -xvf archive.tar -C /tmp
```

Si l'archive était compressée avec bzip2, il faudrait remplacer l'option -z (pour gzip) par **-j** (pour bzip2). Et pour une archive compressée avec xz, il faudrait utiliser **-J** (pour xz).

- Lister le contenu d'une archive

```
tar -tvf archive.tar
```

- Extraire seulement une partie de l'archive. Si votre archive contient par exemple le répertoire rep, vous pouvez n'extraire que ce répertoire :

```
tar -xvf archive.tar rep
```

2 La compression des données gzip

La commande **gzip** (Gnu ZIP) permet de compresser des fichiers pour réduire la place qu'ils prennent sur le disque. Cette compression est souvent utilisée avant le transfert du fichier à travers un réseau de communication pour réduire le temps de transfert.

La commande gzip s'utilise de la manière suivante :

```
gzip monFichier
```

Le résultat de cette commande est la création d'un fichier *monFichier.gz* et la suppression du fichier *monFichier*. Il est possible de donner plusieurs noms de fichiers ou de compresser récursivement un répertoire sans en changer la hiérarchie. Il faut aussi noter que compresser plusieurs fois un fichier ou compresser un fichier déjà compressé ne permet pas de gagner plus de place sur le disque. Il est ainsi inutile de compresser un fichier.gz, .mp3, .jpg, ...car la taille économisée ne sera pas significative.

La commande de compression **gzip** s'utilise le plus couramment en combinaison avec la commande **tar**. La commande tar transforme une hiérarchie en un seul fichier. Ce fichier est une archive dont le nom est suffixé par l'extension.tar comme par exemple archive.tar. Puis cette archive est compressée avec la commande gzip pour obtenir le fichier suffixé par l'extension **.tar.gz** comme dans notre exemple **archive.tar.gz**. Ce dernier fichier est parfois renommé **archive.tgz** pour que le nom ne comporte qu'un seul point et qu'une seule extension.

Pour conserver le fichier original et en même temps produire un fichier compressé, on utilise l'option **-c** de la commande gzip. Attention au fait que cette option ne crée plus automatiquement le fichier résultat, mais redirige les données compressées vers la sortie standard. Aussi, il faut rediriger cette sortie vers le fichier à produire :

```
gzip -c monFichier > monFichier.gz
```

La commande précédente permet de conserver le fichier ***monFichier*** et de créer un fichier ***monFichier.gz***.

1 tar et gzip

La commande gzip est aussi une option de la commande tar. Ainsi dans la commande

```
tar -czvf f.tgz f/
```

l'option **z** permet de compresser directement l'archive créée. Dans notre cas, l'archive du répertoire f/ sera compressée et sera sous le fichier **f .tgz**. L'extraction se demande avec l'option x.

2 La commande gunzip

La commande gunzip (GNU unzip) permet de décompresser des fichiers d'extension.gz. Ainsi, la commande gunzip va créer un fichier monFichier et détruire le fichier monFichier.gz.

```
gunzip monFichier.gz
```

Pour conserver le fichier original et en même temps produire le contenu du fichier décompressé sur la sortie standard, on utilise l'option -c, comme pour la commande gzip. Avec cette option il faut rediriger la sortie standard vers le nom du fichier à produire :

```
gunzip -c monFichier.gz > monFichier
```

La commande précédente permet de conserver le fichier ***monFichier.gz*** et de créer un fichier ***monFichier***.

3 Transférer les données

1 La commande scp

Dans les systèmes Unix, il est possible de copier des fichiers entre deux systèmes distants. La commande **scp** utilise le protocole SSH pour transférer les données. Cette commande nécessite un nom d'utilisateur et un mot de passe sur les deux systèmes (source et destination). Ces noms d'utilisateurs et mots de passe peuvent être différents.

Contrairement à d'autres méthodes permettant de copier des fichiers depuis ou vers un système distant au travers du réseau, la commande scp sécurise le transfert avec l'utilisation de SSH, il sera donc impossible à une personne capturant le transfert sur le réseau de reconstituer le contenu du transfert.

La syntaxe et les options de la commande ***scp*** sont très proches de celles de la commande ***cp***.

```
scp alice@src:RepSrc/f1 bob@dest:RepDst/f2
```

La commande précédente permettra de copier le fichier ***RepSrc/f1*** se trouvant sur l'ordinateur ***src*** qui possède une utilisatrice ***alice*** vers un fichier ***RepDst/f2*** se trouvant sur l'ordinateur ***dest*** qui possède un utilisateur ***bob***. Les valeurs de ***src*** et de ***dest*** peuvent par exemple être les adresses IP des ordinateurs ou leur nom de domaine. Pour copier

des fichiers d'un hôte distant à un autre, vous devrez entrer les mots de passe des deux comptes après avoir exécuté cette commande dans votre terminal.

2 Fichier local vers une destination distante

L'utilisation la plus courante de la commande ***scp*** est la copie d'un fichier se trouvant sur l'ordinateur local vers un ordinateur distant.

```
scp /home/alice/scp.zip bob@dest:/home/bob/scp.zip
```

La commande précédente permet de copier le *fichier/home/alice/scp.zip* (fichier local) vers le fichier *distant/home/bob/scp.zip* de l'utilisateur bob se trouvant sur la machine *dest*. Pour identifier une machine sur le réseau, on utilise souvent une adresse.

Si vous voulez copier un dossier qui contient plus de fichiers et des sous-dossiers, utilisez l'option **-r**

3 Fichier distant vers une machine locale

Dans ce processus, la source et la cible de la commande sont inversées, ce qui doit donc se refléter dans la syntaxe. Cette fois, nous copions scp.zip du même hôte distant vers notre machine locale :

```
scp bob@dest:/home/bob/scp.zip /home/alice/scp.zip
```

4 La commande wget

Contrairement à la commande **scp**, la commande **wget** est uni-directionnelle. C'est-à-dire qu'elle ne permet que de copier un fichier se trouvant sur un ordinateur distant vers un l'ordinateur local. La commande wget utilise le plus souvent le protocole HTTP bien qu'il puisse communiquer aussi au travers des protocoles comme FTP. La commande ***wget*** permet donc de copier localement un fichier se trouvant sur un serveur web. L'utilisation la plus courante de la commande wget est le téléchargement d'archives conservées sur un serveur web.

```
wget http://www.example.com/archive.tgz
```

La commande précédente permet de télécharger dans le répertoire courant (répertoire depuis lequel la commande wget est lancée) l'archive archive.tgz se trouvant sur le site web http://www.example.com/. À la fin de l'exécution correcte de la commande, une copie du fichier archive.tgz se trouvera sur la machine locale. L'option **-c** de la commande ***wget*** permet la reprise d'un téléchargement.

5 Transfert de fichier avec sftp

Le protocole de transfert de fichiers (FTP) était un protocole largement utilisé pour transférer des fichiers, cependant ce n'était pas un moyen sécurisé pour transférer car les données n'étaient pas cryptées.

SFTP signifie : **SSH File Transfer Protocol** ou **Secure File Transfer Protocol**. Ce protocole est une extension du protocole SSH. Il permet d'effectuer le transfert de fichiers. On pourrait parler de transfert FTP encapsulé dans un tunnel SSH sécurisé.
Il utilise le port SSH (22), pour le transfert des données ainsi que pour le contrôle.

Pour démarrer une session **sftp**, à l'invite de commande, entrez le nom d'utilisateur et le nom d'hôte distant .

```
pi@raspberrypi:~ $ sftp pi@192.168.1.10
The authenticity of host '192.168.1.10 (192.168.1.10)' can't be established.
ECDSA key fingerprint is SHA256:toIUcxbLl6jMLag4HdZDmsBXeuexAuz31FF73lMfajI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.10' (ECDSA) to the list of known hosts.
pi@192.168.1.10's password:
Connected to pi@192.168.1.10.
sftp>
```

Une fois l'authentification réussie, vous verrez un shell avec une invite **sftp>**

vérifiez les commandes disponibles en tapant '?' ou 'help' à l'invite de commande.

```
sftp> help
Available commands:
bye                                Quit sftp
cd path                            Change remote directory to 'path'
chgrp grp path                     Change group of file 'path' to 'grp'
chmod mode path                    Change permissions of file 'path' to 'mode'
chown own path                     Change owner of file 'path' to 'own'
df [-hi] [path]                    Display statistics for current directory or
                                   filesystem containing 'path'
exit                               Quit sftp
get [-afPpRr] remote [local]       Download file
reget [-fPpRr] remote [local]      Resume download file
reput [-fPpRr] [local] remote      Resume upload file
help                               Display this help text
lcd path                           Change local directory to 'path'
lls [ls-options [path]]            Display local directory listing
lmkdir path                        Create local directory
ln [-s] oldpath newpath            Link remote file (-s for symlink)
lpwd                               Print local working directory
```

```
ls [-1afhlnrSt] [path]             Display remote directory listing
lumask umask                       Set local umask to 'umask'
mkdir path                         Create remote directory
progress                           Toggle display of progress meter
put [-afPpRr] local [remote]       Upload file
pwd                                Display remote working directory
quit                               Quit sftp
rename oldpath newpath             Rename remote file
rm path                            Delete remote file
rmdir path                         Remove remote directory
symlink oldpath newpath            Symlink remote file
version                            Show SFTP version
!command                           Execute 'command' in local shell
!                                  Escape to local shell
?                                  Synonym for help
sftp>
```

La commande '**lpwd**' est utilisée pour vérifier le répertoire de travail actuel local, tandis que la commande '**pwd**' est utilisée pour vérifier le répertoire de travail distant.

```
sftp> pwd
Remote working directory: /home/pi
sftp> lpwd
Local working directory: /home/pi
sftp>
```

Lister les fichiers et les répertoires dans le système local ***lls*** et distant. *ls*

```
sftp> ls
Desktop             Documents           Downloads
Music               Pictures            Public              Ruche
Templates           Videos              apprendre_cpp
oldconffiles
python_games        raspberry_pi3_www   test
sftp> lls
ABC_du_C       Desktop    letsencrypt  Music     python     vendor
composer.json  Documents  linux        Pictures  Templates  Videos
composer.lock  Downloads  MagPi        Public    toto
sftp>
```

Transferer des fichiers

Transfert montant d’un fichier (upload)

```
sftp> put toto
```

```
Uploading toto to /home/pi/toto
toto                                          100%  408    44.7KB/s   00:00
sftp>
```

Transfert montant de multiples fichiers

```
sftp> mput Mag*
Uploading MagPi85.pdf to /home/pi/MagPi85.pdf
MagPi85.pdf                                   100%   35MB   1.9MB/s   00:18
sftp>
```

Transfert descendant téléchargement d'un fichier (download)

```
sftp> get toto
Fetching /home/pi/toto to toto
/home/pi/toto                         100%  408    13.9KB/s   00:00
```

Transfert descendant téléchargement de multiples fichiers

```
sftp> mget *.txt
```

Comme nous pouvons le voir, par défaut, la commande ***get***, télécharge le fichier dans le système local avec le même nom. Nous pouvons télécharger un fichier distant avec un nom différent en précisant le nom à la fin. (Cela s'applique uniquement lors du téléchargement d'un seul fichier).

Quiter sftp

```
sftp> quit
quit
pi@raspberrypi:~ $
```

6 Travailler à distance

Il existe plusieurs commandes permettant d'effectuer du travail à distance, c'est-à-dire d'ouvrir une session interactive sur une machine distante avec un shell complet (en incluant toutes les commandes de Bash). Historiquement, la commande ***telnet*** était utilisée pour le travail à distance. Elle n'est cependant plus utilisée à cause de problèmes de sécurité. En effet, la commande telnet n'encrypte pas les échanges. La commande utilisée actuellement est la commande ***ssh***.

La commande ssh (Secure Shell) permet d'ouvrir une session interactive sur une machine distante mais nécessite d'avoir un nom d'utilisateur et un mot de passe sur cette machine distante. Pour se connecter à la machine exemple.com avec le nom d'utilisateur alice nous utilisons la commande

```
ssh alice@exemple.com
```