

[Accueil](#) > [Cours](#) > [Reprenez le contrôle à l'aide de Linux !](#) > Transférer des fichiers

Reprenez le contrôle à l'aide de Linux !

🕒 30 heures



Facile

Mis à jour le 29/06/2021



Transférer des fichiers

Vous avez appris à vous connecter à un serveur à distance avec SSH. Désormais, grâce au réseau, vous pouvez exécuter des commandes sur un ordinateur en prenant le contrôle à distance.

On continue ici notre découverte du monde fabuleux des réseaux sous Linux. Un monde un peu particulier, comme vous avez pu le découvrir : de gros efforts sont faits pour assurer la sécurité des données grâce au cryptage, ce qui permet d'éviter de se faire voler ses données personnelles, comme son mot de passe par exemple.

Dans ce chapitre, nous allons mettre l'accent sur le transfert de fichiers : comment télécharger un fichier ? Comment se connecter à un FTP, lire et télécharger des fichiers ? Et surtout, comment copier des fichiers de manière sécurisée ?

wget : téléchargement de fichiers



Nous commençons par une commande simple à utiliser, du moins en apparence : `wget`. Elle permet de télécharger des fichiers directement depuis la console.

Il suffit d'indiquer l'adresse HTTP ou FTP d'un fichier à télécharger :

```
$ wget http://cdimage.debian.org/debian-cd/4.0_r5/i386/iso-cd/ debian-40r5-i386-businesscard.iso
```

Une barre de progression du téléchargement devrait alors s'afficher dans la console :

```
$ wget http://cdimage.debian.org/debian-cd/4.0_r5/i386/iso-cd/ debian-40r5-i386-businesscard.iso
--2008-12-05 12:43:25-- http://cdimage.debian.org/debian-cd/4.0_r5/ i386/iso-cd/debian-40r5-i386-businesscard.iso
Résolution de cdimage.debian.org... 130.239.18.173, 130.239.18.137
Connexion vers cdimage.debian.org[130.239.18.173]:80... connecté.
requête HTTP transmise, en attente de la réponse... 302 Found
Emplacement: http://saimei.acc.umu.se/debian-cd/4.0_r5/i386/iso-cd/ debian-40r5-i386-businesscard.iso [suivant]
--2008-12-05 12:43:25-- http://saimei.acc.umu.se/debian-cd/4.0_r5/ i386/iso-cd/debian-40r5-i386-businesscard.iso
Résolution de saimei.acc.umu.se... 130.239.18.138
Connexion vers saimei.acc.umu.se[130.239.18.138]:80... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Longueur: 34181120 (33M) [application/octet-stream]
Saving to: `debian-40r5-i386-businesscard.iso'

38% [=====] 13 208 331 117K/s eta 70s
```

Les informations au début sont assez nombreuses et ne nous intéressent pas vraiment. Elles indiquent simplement comment le programme a communiqué avec le serveur qui possédait le fichier.

En bas, vous avez, dans l'ordre et sur la dernière ligne :

1. une barre de progression qui se met à jour ;
2. le nombre d'octets téléchargés ;
3. la vitesse de téléchargement ;
4. le temps restant estimé (`eta`).

Vous pouvez arrêter le téléchargement à tout moment en utilisant la combinaison `Ctrl + C` que vous avez déjà découverte.



Comment puis-je récupérer l'adresse du fichier à télécharger pour la donner à `wget` ?

Le plus simple est d'ouvrir un navigateur web tel que Firefox là où vous avez accès à un environnement graphique et de faire un clic droit sur le lien du fichier que vous voulez télécharger, pour enfin sélectionner « Copier l'adresse du lien ». Vous pouvez ensuite le coller dans la console.

Notez qu'il existe aussi des navigateurs en console tels que `lynx` (plutôt basique) et `links` (assez complet) que vous pouvez télécharger et essayer si vous le souhaitez.

Reprendre un téléchargement arrêté

Si vous voulez reprendre un téléchargement arrêté, utilisez l'option `-c` :

```
$ wget -c http://cdimage.debian.org/debian-cd/4.0_r5/i386/iso-cd/ debian-40r5-i386-businesscard.iso
```

Pour que cela fonctionne, il ne faut bien évidemment pas supprimer le bout de fichier téléchargé sur votre disque. :-)

Si la reprise a fonctionné vous devriez voir une barre de progression comme celle-ci :

```
71% [+++++++=====>] 24 450 216 470K/s eta 88s
```


Les `+++` correspondent à la partie précédemment téléchargée. Cela vous confirme que la reprise a bien fonctionné.

Lancer un téléchargement en tâche de fond

Enfin, si vous voulez que le téléchargement soit envoyé en tâche de fond dès le début, il y a la technique du `nohup` que l'on connaît et qui s'applique à toutes les commandes, mais vous pouvez aussi utiliser l'option `--background` :

```
$ wget --background -c http://cdimage.debian.org/debian-cd/4.0_r5/ i386/iso-cd/debian-40r5-i386-businesscard.iso
Poursuite à l'arrière plan, pid 8422.
La sortie sera écrite vers « wget-log ».
```

Comme indiqué, l'avancement du téléchargement sera écrit dans un fichier `wget-log` .




`wget` propose une quantité impressionnante d'options et je ne peux donc pas toutes les traiter. Sachez que vous pouvez notamment l'utiliser pour télécharger des pages web. Consultez le manuel pour en savoir plus car il y a de quoi faire : `man wget` .

Un des avantages de `wget` est que vous avez toujours une barre de progression, et cela même si vous téléchargez un fichier depuis un serveur FTP. Ça paraît bête, mais, contrairement à `wget` , le programme `ftp` que l'on verra plus loin ne donne pas l'avancement du téléchargement !

scp : copier des fichiers sur le réseau

Vous connaissez la commande `cp` ? Elle permet de copier des fichiers sur votre disque dur.

Eh bien voici `scp` (*Secure CoPy*), qui permet de copier des fichiers d'un ordinateur à un autre à travers le réseau ! Le tout de manière sécurisée, bien sûr. :-)



Il existe aussi `rcp` (*Remote CoPy*) qui fait la même chose mais sans aucun cryptage. Son utilisation est déconseillée.


`scp` s'utilise quasiment comme `ssh` . D'ailleurs ce n'est pas un hasard car `scp` s'appuie sur `ssh` pour fonctionner. Là où `ssh` sert à ouvrir une console à distance (un shell), `scp` est spécialement conçue pour copier des fichiers d'un ordinateur à un autre.

On l'utilise comme ceci :

```
scp fichier_origine copie_destination
```

Le premier élément à indiquer est la position du fichier que l'on veut copier. Le second élément correspond au répertoire de destination où il doit être copié.

Chacun de ces éléments peut s'écrire sous la forme suivante : `login@ip:nom_fichier` . Le login et l'IP sont facultatifs. Si vous n'écrivez ni login ni IP, `scp` considérera que le fichier se trouve sur votre ordinateur.



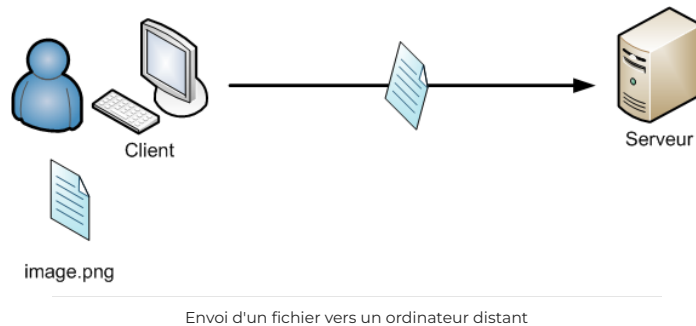
Vous pouvez remplacer l'IP par un nom d'hôte (un nom de domaine) si vous en avez un qui est plus facile à retenir. Par exemple, le serveur Lisa du Site du Zéro peut être appelé en écrivant `lisa.simple-it.fr` au lieu d'une IP compliquée comme `85.123.10.201` .

Copier un fichier de votre ordinateur vers un autre

Prenons un cas concret (figure suivante) pour que vous puissiez mieux voir comment ça s'utilise :

```
scp image.png mateo21@85.123.10.201:/home/mateo21/images/
```

Ici, je demande à copier le fichier `image.png` qui se trouve sur mon ordinateur vers un autre ordinateur dont l'IP est `85.123.10.201` . Sur cet autre ordinateur, le fichier sera placé dans le dossier `/home/mateo21/images/` .



Notez que l'on peut utiliser le symbole `~` pour indiquer « mon répertoire personnel » (`/home/mateo21/`). D'autre part, si cet autre ordinateur a un nom d'hôte facile à retenir, j'aurais tendance à l'utiliser à la place de l'IP. J'aurais donc pu écrire quelque chose comme ce qui suit et qui aurait été identique :

```
scp image.png mateo21@lisa.simple-it.fr:~/images/
```

Lorsque vous lancez la commande, `scp` essaiera de se connecter au serveur ayant l'IP indiquée avec le login que vous avez demandé (`mateo21` , dans mon cas). On vous demandera alors votre mot de passe ou, mieux, `scp` utilisera votre clé privée si elle existe.

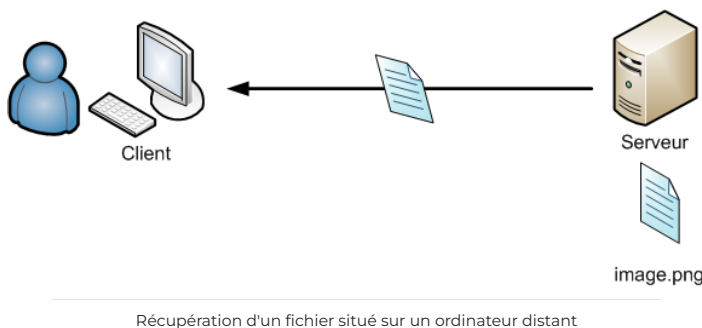
Nous avons vu que, combiné à l'agent SSH, cela nous évitait d'avoir à retaper notre `passphrase` à chaque fois !

Copier un fichier d'un autre ordinateur vers le vôtre

On peut également faire le contraire (figure suivante), c'est-à-dire récupérer un fichier qui se trouve sur un autre ordinateur et le placer chez nous :

```
scp mateo21@85.123.10.201:image.png copie_image_sur_mon_pc.png
```

Dans cet exemple, je copie le fichier `image.png` qui se trouve sur le serveur dont l'IP est `85.123.10.201` et place cette copie sur mon propre ordinateur sous le nom `copie_image_sur_mon_pc.png` .



Si je veux, je peux aussi copier le fichier sans en changer le nom :

```
scp mateo21@85.123.10.201:image.png .
```

Notez le point à la fin. Il signifie « copier dans le répertoire dans lequel je me trouve ». Le fichier `image.png` sera donc placé sur mon ordinateur dans le dossier actuel.

Le piège du port

Si le serveur SSH auquel vous essayez de vous connecter n'est pas sur le port standard (22), il faudra indiquer le numéro du port avec l'option `-P` :

```
scp -P 16296 mateo21@85.123.10.201:image.png .
```

La commande `ssh` a aussi une option pour indiquer un port ; vous vous en souvenez peut-être, c'est... `-p` ! Faites donc attention car avec `ssh` c'est un « p » minuscule qu'il faut utiliser alors qu'avec `scp` c'est un « P » majuscule !
Je peux vous dire que je me suis trompé un bon nombre de fois.

ftp & sftp : transférer des fichiers

Le FTP (*File Transfer Protocol*) est un protocole permettant d'échanger des fichiers sur le réseau. Il est assez ancien (1985) et toujours utilisé à l'heure actuelle pour transférer des fichiers.

On l'utilise en général dans deux cas.


- Pour télécharger un fichier depuis un serveur FTP public. En général, les navigateurs web font cela de manière automatique et transparente lorsque vous cliquez sur un lien de téléchargement.

La connexion se fait alors en **mode anonyme**.

- Pour transférer des fichiers vers un serveur FTP privé (et éventuellement en télécharger aussi). Lorsque l'on prend un hébergement pour son site web, l'hébergeur nous donne en général des accès FTP pour aller y déposer les fichiers du site.

La connexion se fait donc en **mode authentifié**.

Tout le monde n'a pas forcément accès à un serveur FTP privé, aussi je vous propose pour les exemples suivants de vous connecter à un serveur FTP public (rassurez-vous, si vous voulez vous connecter à un FTP privé, la méthode reste la même).

 Nous nous intéressons ici au fonctionnement du FTP en ligne de commande. Bien sûr, il existe des logiciels graphiques qui font la même chose, comme par exemple FileZilla.

Connexion à un serveur FTP

Essayons de nous connecter au serveur FTP de Debian, accessible à l'adresse suivante : `ftp://ftp.debian.org`.

```
$ ftp ftp.debian.org
```

Le serveur FTP devrait répondre en vous demandant un login et un mot de passe. Pour les serveurs FTP publics, le login à utiliser est toujours `anonymous` (anonyme).

```
$ ftp ftp.debian.org
Connected to ftp.debian.org.
220 saens.debian.org FTP server (vsftpd)
Name (ftp.debian.org:mateo21): anonymous
331 Please specify the password.
Password:
```

Pour le mot de passe, peu importe ce que vous mettez, vous serez acceptés. :-)

Vous devriez alors voir s'afficher un message de bienvenue se terminant par :

```
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Vous avez maintenant un **prompt** (il s'agit du terme anglais pour *invite de commandes*) `ftp>` qui vous permet de rentrer des commandes FTP.

Se déplacer au sein du serveur FTP

Vous savez quoi ? Bonne nouvelle : les commandes que vous pouvez utiliser sont pour la plupart les mêmes que celles que vous connaissez. :-)

Citons dans le lot :

- `ls` : affiche le contenu du répertoire actuel ;
- `pwd` : affiche le chemin du répertoire actuel ;
- `cd` : change de répertoire.

Avec ces commandes, vous devriez déjà pouvoir vous balader sur le serveur FTP.

Faites un `ls` pour voir :

```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxrwsr-x  7 1176   1176   4096 Dec 05 09:10 debian
226 Directory send OK.
```

Les lignes commençant par un numéro sont des messages envoyés par le serveur FTP. Vous noterez que les fichiers s'affichent comme si l'on avait écrit `ls -l`.

Il y a seulement un répertoire, rendez-vous donc dans `debian` :

```
ftp> cd debian
250 Directory successfully changed.
```

Affichez à nouveau le contenu :

```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-rw-r--  1 1176   1176   940 Oct 27 20:29 README
-rw-rw-r--  1 1176   1176  1290 Dec 04 2000 README.CD-manufacture
```

```

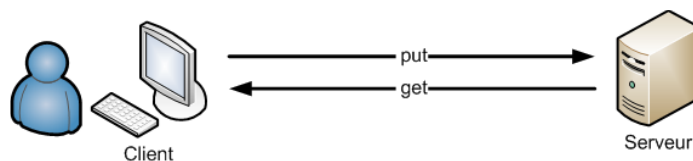
-rw-rw-r-- 1 1176 1176 2426 Oct 27 20:29 README.html
-rw-r--r-- 1 1176 1176 124286 Dec 03 19:52 README.mirrors.html
-rw-r--r-- 1 1176 1176 62059 Dec 03 19:52 README.mirrors.txt
drwxr-sr-x 9 1176 1176 4096 Nov 16 18:56 dists
drwxr-sr-x 3 1176 1176 4096 Nov 11 22:16 doc
drwxr-sr-x 3 1176 1176 4096 Dec 05 09:08 indices
-rw-rw-r-- 1 1176 1176 4557196 Dec 05 08:49 ls-lR.gz
-rw-r--r-- 1 1176 1176 154934 Dec 05 08:49 ls-lR.patch.gz
drwxr-sr-x 5 1176 1176 4096 Nov 11 22:16 pool
drwxr-sr-x 4 1176 1176 4096 Nov 18 09:04 project
226 Directory send OK.

```

Le transfert de fichiers

Si vous souhaitez récupérer un fichier ou en envoyer un, il y a deux commandes à connaître (figure suivante) :

- `put` : envoie un fichier vers le serveur ;
- `get` : télécharge un fichier depuis le serveur.



Notez qu'il est impossible d'utiliser `put` sur les serveurs FTP publics comme celui auquel nous sommes connectés. Seul le téléchargement de fichiers est autorisé.

D'autres commandes, comme `chmod` (qui permet de modifier les droits des fichiers), ne sont pas activées non plus.

Vous pouvez par exemple récupérer le fichier `README` en écrivant `get README` :

```

ftp> get README
local: README remote: README
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for README (940 bytes).
226 File send OK.
940 bytes received in 0.00 secs (918.9 kB/s)

```

Et voilà ! Le fichier se trouve maintenant sur votre ordinateur.

Il a été téléchargé dans le dossier dans lequel vous vous trouviez sur votre ordinateur. Pour savoir dans quel dossier vous êtes **chez vous**, tapez `!pwd` :

```

ftp> !pwd
/home/mateo21

```

Si vous voulez changer de dossier **chez vous**, utilisez `!cd`. Pour lister les fichiers **chez vous**, utilisez `!ls`. Bref, vous m'avez compris, il suffit de faire précéder les commandes d'un point d'exclamation pour qu'elles s'exécutent chez vous et non sur le serveur FTP.

Les autres commandes

Il existe de nombreuses autres commandes FTP, nous n'allons pas toutes les voir.

Tapez `man ftp` pour obtenir un aperçu des commandes disponibles. Vous noterez que toutes ne sont pas identiques à celles que vous connaissez. Par exemple, pour supprimer un fichier, ce n'est pas `rm` mais... `delete` ! D'autres vous seront familières : `mkdir` permet de créer un dossier, par exemple.

Pour quitter le serveur et vous déconnecter, vous avez le choix entre la bonne vieille combinaison de touches `Ctrl + D` qui commande la fermeture de la session ou encore les commandes `bye`, `exit` et `quit`, qui sont analogues.

sftp : un FTP sécurisé

Le protocole FTP a un défaut : il n'est pas sécurisé ; les données ne sont pas cryptées. Quelqu'un ayant accès au réseau pourrait alors intercepter le contenu des fichiers que vous échangez ou encore votre mot de passe lors de la connexion.

Pour remédier à cela, on a inventé `sftp`, qui repose sur SSH pour sécuriser la connexion :

```
sftp login@ip
```

Par exemple :

```
sftp mateo21@lisa.simple-it.fr
```

On vous demandera alors votre mot de passe (bien entendu, la clé publique sera utilisée, si elle est présente).

Une fois que vous serez connectés, les commandes sont presque les mêmes que pour le FTP. Vous retrouverez notamment `get` et `put` pour échanger des fichiers. Sachez que les commandes sont globalement plus puissantes et pratiques en SFTP qu'en FTP.

Méfiez-vous toutefois, certaines commandes changent ! Par exemple, pour supprimer un fichier, ce n'est plus `delete` mais à nouveau `rm` ! Lisez le manuel pour plus d'informations : `man sftp` .



Pour se connecter en SFTP, on utilise le même port que SSH (soit 22 par défaut). Si votre serveur SSH fonctionne sur un autre port, vous devrez le préciser comme ceci : `sftp -oPort=27401 mateo21@serveur` .

À l'heure actuelle, le SFTP reste assez peu utilisé. Les hébergeurs web utilisent toujours le FTP classique alors que la plupart des logiciels graphiques comme FileZilla sont pourtant capables de se connecter en SFTP.

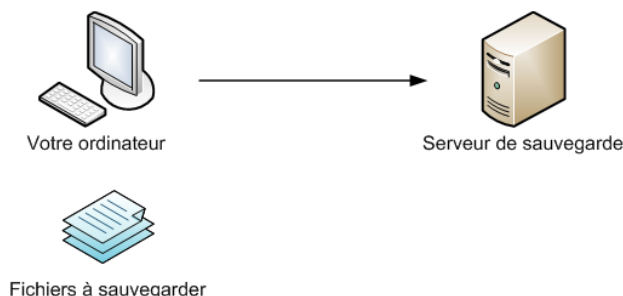
rsync : synchroniser des fichiers pour une sauvegarde



`rsync` est un programme assez simple à utiliser et pourtant très puissant. Il permet d'effectuer une synchronisation entre deux répertoires, que ce soit sur le même PC ou entre deux ordinateurs reliés en réseau.

`rsync` est le plus souvent utilisé pour effectuer des **sauvegardes incrémentielles**. Je m'explique.

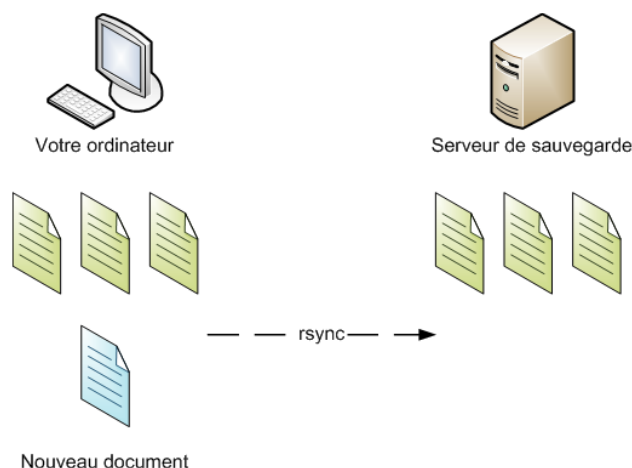
Par exemple, imaginez que vous souhaitiez sauvegarder régulièrement les fichiers de votre home (répertoire personnel). Ainsi, si un jour un accident survient (perte du PC, par exemple), vous aurez toujours vos documents sauvegardés au chaud sur un serveur quelque part (figure suivante).



D'accord, vous copiez tout votre home sur ce serveur.

Oui mais voilà, votre home contient peut-être 10 Go de fichiers : entre vos vidéos, la musique, vos rapports au format texte et vos photos de vacances, ça va vite.

Sauvegarder ces 10 Go une fois, d'accord. Mais la seconde fois, plutôt que de tout renvoyer, vous souhaitez peut-être envoyer uniquement les fichiers qui ont été ajoutés ou modifiés, non ?



C'est là que `rsync` intervient. C'est une sorte de `scp` intelligent : il compare et analyse les différences entre deux dossiers puis copie uniquement les changements. C'est ce que veut dire le mot « incrémentiel ».

`rsync` peut être utilisé pour effectuer une sauvegarde entre deux dossiers sur le même ordinateur ou bien entre deux dossiers sur deux ordinateurs différents (figure suivante). En général, on l'utilise plutôt pour sauvegarder entre deux ordinateurs différents, bien sûr.

Dans un premier temps, pour faire simple, nous allons voir comment fonctionne la sauvegarde entre deux dossiers de votre ordinateur puis nous effectuerons ensuite la sauvegarde sur un autre PC.

Sauvegarder dans un autre dossier du même ordinateur

Dans les exemples qui vont suivre, je vais supposer que vous souhaitez sauvegarder le dossier `Images` dans un dossier `backups` .

Dans le dossier `Images`, il y a quelques photos de vacances :

```
$ ls
espagne1.jpg  italie1.jpg  italie2.jpg  italie3.jpg
```

Vous pouvez créer comme moi des fichiers de test à l'aide de la commande `touch`.

Maintenant, lancez un `rsync` comme ceci :

```
$ rsync -arv Images/ backups/
sending incremental file list
created directory backups
./
espagne1.jpg
italie1.jpg
italie2.jpg
italie3.jpg

sent 268 bytes  received 91 bytes  718.00 bytes/sec
total size is 0  speedup is 0.00
```

Quelques explications concernant les paramètres :

- `-a` : conserve toutes les informations sur les fichiers, comme les droits (`chmod`), la date de modification, etc. ;
- `-r` : sauvegarde aussi tous les sous-dossiers qui se trouvent dans le dossier à sauvegarder ;
- `-v` : mode verbeux, affiche des informations détaillées sur la copie en cours.

Viennent ensuite le nom du dossier à sauvegarder et le répertoire de sauvegarde.

`rsync` analyse le contenu du répertoire de sauvegarde dans un premier temps. Comme celui-ci est vide, vous pouvez constater qu'il y a copié tous les fichiers.

Maintenant, lancez la même commande une seconde fois :

```
$ rsync -arv Images/ backups/
sending incremental file list

sent 109 bytes  received 12 bytes  242.00 bytes/sec
total size is 0  speedup is 0.00
```

Comme vous pouvez le voir, cette fois aucun fichier n'a été envoyé ! En effet, `rsync` étant intelligent, il a détecté qu'il n'y avait aucun changement et donc qu'il n'y avait pas lieu de copier quoi que ce soit.

Testons un peu ce qui se passe si l'on ajoute un fichier :

```
$ touch Images/espagne2.jpg
$ rsync -arv Images/ backups/
sending incremental file list
./
espagne2.jpg

sent 172 bytes  received 34 bytes  412.00 bytes/sec
total size is 0  speedup is 0.00
```

Le nouveau fichier `espagne2.jpg` a bien été copié ! ;-)

Vous pouvez aussi essayer de modifier un fichier, vous verrez que `rsync` copie bien ceux qui ont été modifiés.

Supprimer les fichiers



J'ai essayé de supprimer un fichier mais celui-ci n'a pas été supprimé dans le répertoire de sauvegarde. Comment faire ?

Par défaut, `rsync` ne supprime pas les fichiers dans le répertoire de copie. Si vous voulez lui demander de le faire, pour que le contenu soit strictement identique, rajoutez `--delete`.

Par exemple, si je supprime le fichier `italie3.jpg` :

```
$ rm Images/italie3.jpg
$ rsync -arv --delete Images/ backups/
sending incremental file list
deleting italie3.jpg

sent 120 bytes  received 12 bytes  264.00 bytes/sec
total size is 4  speedup is 0.03
```

... `rsync` me supprime mon fichier `italie3.jpg` !

Sauvegarder les fichiers supprimés

Peut-être que la suppression du fichier était accidentelle. Si même votre `rsync` supprime le fichier dans le répertoire de sauvegarde, vous n'en aurez plus aucune trace !

Heureusement, il est possible de garder de côté les fichiers que l'on a supprimés, sait-on jamais, au cas où...

Pour cela, rajoutez l'option `--backup`. Les fichiers supprimés prendront un suffixe dans le répertoire de sauvegarde.

Vous pouvez aussi, pour éviter que ça ne fasse désordre, déplacer les fichiers supprimés dans un dossier qui leur est dédié. Rajoutez

```
--backup-dir=/chemin/vers/le/repertoire .
```

Par exemple :

```
$ rsync -arv --delete --backup --backup-dir=/home/mateo21/backups_supprimes Images/ backups/
```

! Je vous recommande d'indiquer le répertoire `backup-dir` en absolu comme je l'ai fait. Sinon, le répertoire des fichiers supprimés sera placé à l'intérieur du répertoire de sauvegarde et vous risquez d'avoir plus de problèmes qu'autre chose lors de la synchronisation.

`rsync` peut faire bien d'autres choses, comme exclure un dossier de la sauvegarde (option `--exclude`). Je vous laisse lire le manuel pour savoir un peu tout ce que vous pouvez faire.

Sauvegarder sur un autre ordinateur

Intéressons-nous maintenant à la sauvegarde sur un autre ordinateur, parce que là c'est bien joli mais on se sentirait plus en sécurité si les fichiers étaient envoyés ailleurs, sur un autre ordinateur.

L'avantage de `rsync` est qu'il peut copier les fichiers en employant plusieurs méthodes différentes. La plus couramment utilisée, que nous allons choisir ici, est de passer par SSH. Comme quoi, vous le voyez, SSH sert à sécuriser tous types de transferts.

```
$ rsync -arv --delete --backup --backup-dir=/home/mateo21/fichiers_supprimes Images/ mateo21@IP_du_serveur:mes_backups/
```

Si votre serveur SSH écoute sur un autre port que celui par défaut, il faudra rajouter `-e "ssh -p port"` :

```
$ rsync -arv --delete --backup --backup-dir=/home/mateo21/fichiers_supprimes Images/ mateo21@IP_du_serveur:mes_backups/ -e "ssh -p 12473"
```

En résumé

- `wget` permet de télécharger un fichier.
- Pour copier des fichiers d'un ordinateur à un autre, on utilise `scp`. Il fonctionne à l'aide de SSH, donc le transfert est sécurisé.
- On peut se connecter à un serveur FTP avec la commande `ftp` pour y télécharger et y envoyer des fichiers.
- Il existe une alternative sécurisée à FTP qui crypte les échanges grâce à SSH : `sftp`.
- `rsync` permet de synchroniser le contenu de deux dossiers sur un même ordinateur ou sur deux ordinateurs différents. Il est particulièrement utile pour effectuer des sauvegardes.

 **Le professeur** vous de ce cours ?



Mathieu Nebra

Entrepreneur à plein temps, auteur à plein temps et co-fondateur d'OpenClassrooms

Reprenez le contrôle de votre vie numérique AU SUIVANT

Découvrez aussi ce cours en... **SAUVEGARDE À DISTANCE AVEC SSH**

ANALYSER LE RÉSEAU ET FILTRER LE TRAFIC AVEC UN PARE-FEU ➔



Livre



PDF

OPENCLASSROOMS

Qui sommes-nous ?

Financements

Expérience de formation


Forum

Blog 

Presse 

OPPORTUNITÉS

Nous rejoindre 

Devenir mentor 

Devenir coach carrière 

AIDE



FAQ

POUR LES ENTREPRISES

Former et recruter

EN PLUS

Boutique 

Mentions légales

Conditions générales d'utilisation

Politique de protection des données personnelles

Cookies

Accessibilité

 Français 

