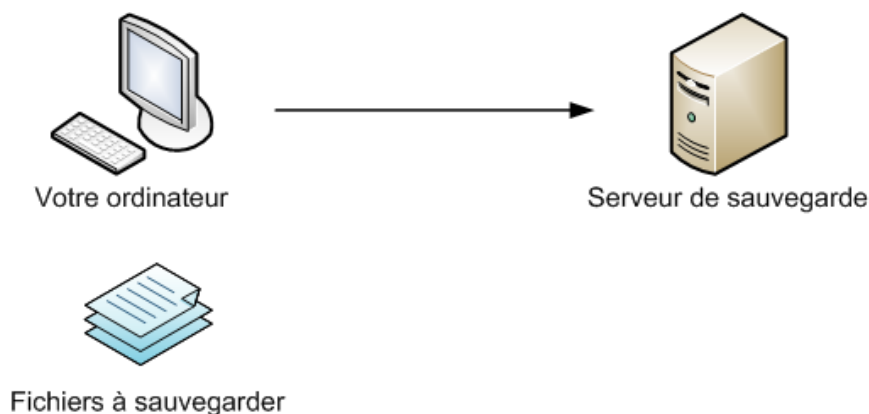


## rsync : synchroniser des fichiers pour une sauvegarde

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.

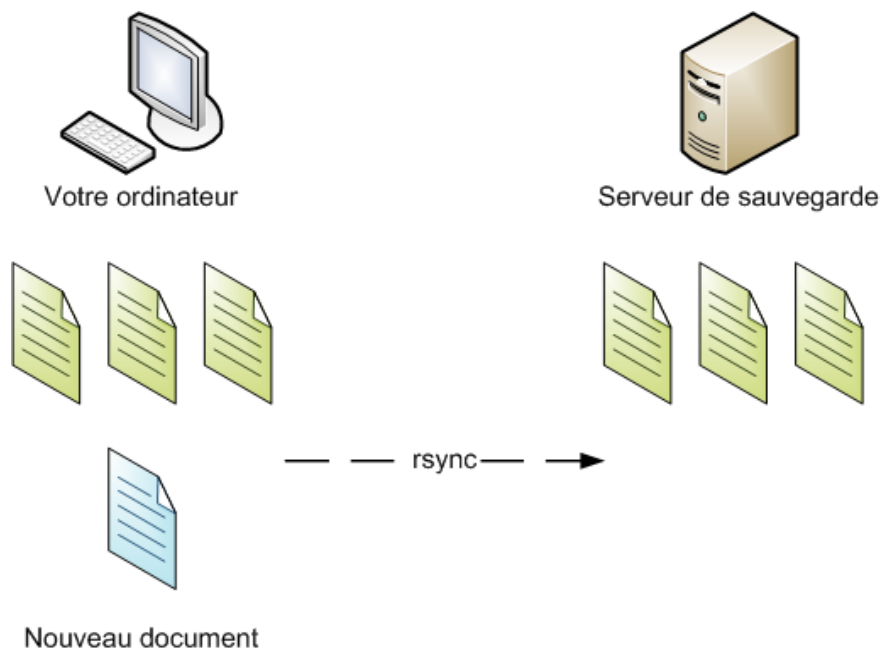
Par exemple, imaginez que vous souhaitez sauvegarder régulièrement les fichiers de votre home (répertoire personnel). Ainsi, si un jour un accident survient (perte du PC, par exemple), vous aurez toujours vos documents sauvegardés au chaud sur un serveur quelque part (figure suivante).



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.