
Sauvegardes et restaurations

La sauvegarde va permettre de répondre à un besoin de conservation et de restauration des données de manière sûre et efficace.

La sauvegarde permet de se protéger des éléments suivants :

- **Destruction** : Volontaire ou involontaire. Humaine ou technique. Virus, ...
- **Suppression** : Volontaire ou involontaire. Humaine ou technique. Virus, ...
- **Intégrité** : Données devenues inutilisables.

Aucun système n'est infaillible, aucun humain n'est infaillible, aussi pour éviter de perdre des données, il faut les sauvegarder pour être en mesure de les restaurer suite à un problème.

Les supports de sauvegarde sont conservés dans une autre pièce (voire bâtiment) que le serveur afin qu'un sinistre ne vienne pas détruire le serveur et les sauvegardes.

De plus, l'administrateur devra régulièrement vérifier que les supports soient toujours lisibles.

1. Généralités

Il existe deux principes, la **sauvegarde** et l'**archive**.

- L'archive détruit la source d'information après l'opération.
- La sauvegarde conserve la source d'information après l'opération.

Ces opérations consistent à enregistrer des informations dans un fichier, sur un périphérique ou un support (bandes, disques, ...).

1.1. La démarche

La sauvegarde nécessite de l'administrateur du système beaucoup de discipline et une grande rigueur. Il est nécessaire de se poser les questions suivantes :

- Quel est le support approprié ?

- Que faut-il sauvegarder ?
- Nombre d'exemplaires ?
- Durée de la sauvegarde ?
- Méthode ?
- Fréquence ?
- Automatique ou manuelle ?
- Où la stocker ?
- Délai de conservation ?

1.2. Méthodes de sauvegardes

- **Complète** : un ou plusieurs **systèmes de fichiers** sont sauvegardés (noyau, données, utilitaires, ...).
- **Partielle** : un ou plusieurs **fichiers** sont sauvegardés (configurations, répertoires, ...).
- **Incrémentale** : uniquement les **fichiers modifiés** depuis la dernière sauvegarde sont sauvegardés.

1.3. Périodicité

- **Ponctuelle** : à un instant donné (avant une mise à jour du système, ...).
- **Périodique** : Journalière, hebdomadaire, mensuelle, ...



Avant une modification du système, il peut être utile de faire une sauvegarde. Cependant, il ne sert à rien de sauvegarder tous les jours des données qui ne sont modifiées que tous les mois.

1.4. Méthodes de restauration

En fonction des utilitaires disponibles, il sera possible d'effectuer plusieurs types de restaurations.

- **Restauration complète** : arborescences, ...
- **Restauration sélective** : partie d'arborescences, fichiers, ...

Il est possible de restaurer la totalité d'une sauvegarde mais il est également possible d'en restaurer uniquement une partie. Toutefois, lors de la restauration d'un répertoire, les fichiers créés après la sauvegarde ne sont pas supprimés.



Pour récupérer un répertoire tel qu'il était au moment de la sauvegarde il convient d'en supprimer complètement le contenu avant de lancer la restauration.

1.5. Les outils

Il existe de nombreux utilitaires pour réaliser les sauvegardes.

- **outils éditeurs** ;
- **outils graphiques** ;
- **outils mode de commande** : **tar**, **cpio**, **pax**, **dd**, **dump**, ...

Les commandes que nous verrons ici sont **tar** et **cpio**.

- **tar** :
 - simple d'utilisation ;
 - permet l'ajout de fichiers à une sauvegarde existante.
- **cpio** :
 - conserve les propriétaires ;
 - groupes, dates et droits ;
 - saute les fichiers endommagés ;
 - système de fichiers complet.



Ces commandes sauvegardent dans un format propriétaire et standardisé.

1.6. Convention de nommage

L'emploi d'une convention de nommage permet de cibler rapidement le contenu d'un fichier de sauvegarde et d'éviter ainsi des restaurations hasardeuses.

- nom du répertoire ;
- utilitaire employé ;
- options utilisées ;
- date.



Le nom de la sauvegarde doit être un nom explicite.



La notion d'extension sous Unix n'existe pas.

1.7. Contenu d'une sauvegarde

Une sauvegarde contient généralement les éléments suivants :

- le fichier ;
- le nom ;
- le propriétaire ;
- la taille ;
- les permissions ;
- date d'accès.



Le numéro d'inode est absent.

1.8. Modes de stockage

Deux modes de stockage se distinguent :

- fichier sur le disque ;
- périphérique.

2. Tape ArchiveR - tar

La commande tar permet la sauvegarde sur plusieurs supports successifs (options multi-volumes).

Il est possible d'extraire tout ou partie d'une sauvegarde.

Tar sauvegarde implicitement en mode relatif même si le chemin des informations à sauvegarder est mentionné en mode absolu.

2.1. Consignes de restauration

Il faut se poser les bonnes questions

- quoi : Partielle ou complète ;
- où : Lieu où les données seront restaurées ;
- comment : Absolu ou relatif.



Avant une restauration, il faut prendre le temps de la réflexion et déterminer la méthode la mieux adaptée afin d'éviter toutes erreurs.

Les restaurations s'effectuent généralement après un problème qui doit être résolu rapidement. Une mauvaise restauration peut dans certains cas aggraver la situation.

2.2. La sauvegarde avec tar

L'utilitaire par défaut pour créer des archives dans les systèmes UNIX est la commande tar. Ces archives peuvent être compressées avec une compression gzip ou bzip.

Tar permet d'extraire aussi bien un seul fichier ou un répertoire d'une archive, visualiser son contenu ou valider son intégrité, etc.

Créer une archive

Créer une archive non-compressée s'effectue avec les clefs cvf :

Syntaxe de la commande tar pour créer une archive.

```
tar c[vf] [support] [fichiers(s)]
```

Exemple :

```
[root]# tar cvf /sauvegardes/home.133.tar
```

Table 1. Clefs principales de la commande tar

| Clef | Description |
|------|--|
| c | Crée une sauvegarde. |
| v | Affiche le nom des fichiers traités. |
| f | Permet d'indiquer le nom de la sauvegarde (support). |



Il n'y a pas de tiret '-' devant les clefs de tar !

Créer une sauvegarde en mode absolu

Syntaxe de la commande tar pour créer une archive en mode absolu.

```
tar c[vf]P [support] [fichiers(s)]
```

Exemple :

```
[root]# tar cvfP /sauvegardes/home.133.P.tar /home
```

| Clef | Description |
|------|--------------------------------------|
| P | Créer une sauvegarde en mode absolu. |



Avec la clef **P**, le chemin des fichiers à sauvegarder doit être renseigné en **absolu**. Si les deux conditions (clef **P** et chemin **absolu**) ne sont pas indiquées, la sauvegarde est en mode relatif.

Créer une archive compressée avec gzip

Créer une archive compressée en gzip s'effectue avec les clefs cvzf :

```
[root]# tar cvzf archive.tar.gz dirname/
```

| Clef | Description |
|------|-----------------------------|
| z | Comprime l'archive en gzip. |



L'extension .tgz est une extension équivalente à .tar.gz



Conserver les clefs 'cvf' ('tvf' ou 'xvf') inchangée pour toutes les manipulations d'archives et simplement ajouter à la fin des clefs celle de compression simplifie la compréhension de la commande (par exemple 'cvfz' ou 'cvfj', etc.).

Créer une archive compressée avec bzip

Créer une archive compressée en bzip s'effectue avec les clefs cvfj :

```
[root]# tar cvfj archive.tar.bz2 dirname/
```

| Clef | Description |
|------|------------------------------|
| j | Comprime l'archive en bzip2. |



Les extensions .tbz et .tb2 sont des extensions équivalentes à .tar.bz2

gzip vs bzip2

bzip2 nécessite plus de temps pour compresser ou décompresser que gzip mais offre des ratios de compression supérieurs.

Extraire (untar) une archive

Extraire une archive *.tar s'effectue avec les clefs xvf :

```
[root]# tar xvf /sauvegardes/etc.133.tar etc/exports
[root]# tar xvfj /sauvegardes/home.133.tar.bz2
[root]# tar xvfP /sauvegardes/etc.133.P.tar
```



Se placer au bon endroit.

Vérifier le contenu de la sauvegarde.

| Clef | Description |
|------|---|
| x | Extrait des fichiers de l'archive, compressée ou non. |

Extraire une archive tar-gzippée (*.tar.gz) s'effectue avec les clefs xvfz

```
[root]# tar xvfz archive.tar.gz
```

Extraire une archive tar-bzippée (*.tar.bz2) s'effectue avec les clefs xvfj

```
[root]# tar xvfj archive.tar.bz2
```

Lister le contenu d'une archive

Visualiser le contenu d'une archive sans l'extraire s'effectue avec les clefs tvf :

```
[root]# tar tvf archive.tar
[root]# tar tvfz archive.tar.gz
[root]# tar tvfj archive.tar.bz2
```

Lorsque le nombre de fichiers contenus dans une archive devient important, il est possible de passer à la commande less le résultat de la commande tar par un pipe ou en utilisant directement la commande less :

```
[root]# tar tvf archive.tar | less
[root]# less archive.tar
```

Extraire uniquement un fichier d'une archive .tar, tar.gz ou tar.bz2

Pour extraire un fichier spécifique d'une archive tar, spécifier le nom du fichier à la fin de la commande tar xvf.

```
[root]# tar xvf archive.tar /path/to/file
```


La commande précédente permet de n'extraire que le fichier file de l'archive archive.tar.

```
[root]# tar xvfz archive.tar.gz /path/to/file
[root]# tar xvfj archive.tar.bz2 /path/to/file
```

Extraire uniquement un dossier d'une archive tar, tar.gz, tar.bz2

Pour n'extraire qu'un seul répertoire (ses sous-répertoires et fichiers inclus) d'une archive, spécifier le nom du répertoire à la fin de la commande tar xvf.

```
[root] tar xvf archive.tar /path/to/dir/
```

Pour extraire plusieurs répertoires, spécifier chacun des noms les uns à la suite des autres :

```
[root] tar xvf archive_file.tar /path/to/dir1/ /path/to/dir2/
[root] tar xvfz archive_file.tar.gz /path/to/dir1/ /path/to/dir2/
[root] tar xvfj archive_file.tar.bz2 /path/to/dir1/ /path/to/dir2/
```

Extraire un groupe de fichiers d'une archive tar, tar.gz, tar.bz2 grâce à des expressions régulières (regex)

Spécifier une regex pour extraire les fichiers correspondants au pattern spécifié.

Par exemple, pour extraire tous les fichiers avec l'extension .conf :

```
[root] tar xvf archive_file.tar --wildcards '*.conf'
```

Clefs :

- --wildcards *.conf correspond aux fichiers avec l'extension .conf.

Ajouter un fichier ou un répertoire à une archive existante

Il est possible d'ajouter des fichiers à une archive existante avec la clef r.

Par exemple, pour ajouter un fichier :

```
[root]# tar rvf archive.tar filetoadd
```

Le fichier filetoadd sera ajouté à l'archive tar existante. Ajouter un répertoire est similaire :

```
[root]# tar rvf archive_name.tar dirtoadd
```



Il n'est pas possible d'ajouter des fichiers ou des dossiers à une archive compressée.

```
[root]# tar rvfz archive.tgz filetoadd
tar: Cannot update compressed archives
Try `tar --help' or `tar --usage' for more
information.
```

Vérifier l'intégrité d'une archive

L'intégrité d'une archive peut être testée avec la clef W au moment de sa création :

```
[root]# tar cvfW file_name.tar dir/
```

La clef W permet également de comparer le contenu d'une archive par rapport au système de fichiers :

```
[root]# tar tvfW file_name.tar
Verify 1/file1
1/file1: Mod time differs
1/file1: Size differs
Verify 1/file2
Verify 1/file3
```

La vérification avec la clef W ne peut pas être effectuée avec une archive compressée. Il faut utiliser la clef d :

```
[root]# tar dfz file_name.tgz
[root]# tar dfj file_name.tar.bz2
```

Estimer la taille d'une archive

La commande suivante estime la taille d'un fichier tar en KB avant de la créer :

```
[root]# tar cf - /directory/to/archive/ | wc -c
20480
[root]# tar czf - /directory/to/archive/ | wc -c
508
[root]# tar cjf - /directory/to/archive/ | wc -c
428
```

Ajout d'éléments à une sauvegarde existante

Syntaxe de la commande tar pour ajouter un élément à une sauvegarde existante.

```
tar {r|A} [clé(s)] [support] [fichiers(s)]
```

Exemple :

```
[root]# tar rvf /sauvegardes/home.133.tar /etc/passwd
```

| Clef | Description |
|------|--|
| r | Ajoute un ou plusieurs fichiers à la fin d'une sauvegarde sur support à accès direct (disque dur). |
| A | Ajoute un ou plusieurs fichiers à la fin d'une sauvegarde sur support à accès séquentiel (bande). |



Si la sauvegarde a été réalisée en mode relatif, ajoutez des fichiers en mode relatif. Si la sauvegarde a été réalisée en mode absolu, ajoutez des fichiers en mode absolu. En mélangeant les modes, vous risquez d'avoir des soucis au moment de la restauration.

Lire le contenu d'une sauvegarde

Syntaxe de la commande tar pour lire le contenu d'une sauvegarde.

```
tar t[clé(s)] [support]
```

Exemple :

```
[root]# tar tvf /sauvegardes/home.133.tar
[root]# tar tvfj /sauvegardes/home.133.tar.bz2
```

| Clef | Description |
|------|--|
| t | Affiche le contenu d'une sauvegarde (compressée ou non). |



Toujours vérifier le contenu d'une sauvegarde.

Table 2. Convention d'écriture de la commande Tar

| Clés | Fichiers | Suffixe |
|-------|----------|----------------|
| cvf | home | home.tar |
| cvfP | /etc | etc.P.tar |
| cvfz | usr | usr.tar.gz |
| cvfj | usr | usr.tar.bz2 |
| cvfPz | /home | home.P.tar.gz |
| cvfPj | /home | home.P.tar.bz2 |

3. CoPy Input Output - cpio

La commande `cpio` permet la sauvegarde sur plusieurs supports successifs sans indiquer d'options.

Il est possible d'extraire tout ou partie d'une sauvegarde.



`cpio` ne permet pas de sauvegarder directement une arborescence. L'arborescence ou fichiers sont donc transmis sous forme de liste à `cpio`.

Il n'y a aucune option, comme pour la commande `tar`, permettant de sauvegarder et de compresser en même temps. Cela s'effectue donc en deux temps : la sauvegarde puis la compression.

Pour effectuer une sauvegarde avec `cpio`, il faut préciser une liste des fichiers à sauvegarder.

Cette liste est fourni avec les commandes `find`, `ls` ou `cat`.

- `find` : parcourt une arborescence, récursif ou non ;
- `ls` : liste un répertoire, récursif ou non ;
- `cat` : lit un fichier contenant les arborescences ou fichiers à sauvegarder.



`ls` ne peut pas être utilisé avec `-l` (détails) ou `-R` (récursif).

Il faut une liste simple de noms.

3.1. Créer une sauvegarde

Syntaxe de la commande `cpio`.

```
[cde de fichiers |] cpio {-o| --create} [-options] [<fic-liste]
[>support]
```

Exemple :

```
[root]# find /etc | cpio -ov > /sauvegardes/etc.cpio
```

Le résultat de la commande **find** est envoyé en entrée de la commande **cpio** par l'intermédiaire du signe "|" (kbd:[AltGr +6]). Ici, la commande `find /etc` renvoie une liste de fichiers correspondant au contenu du répertoire `/etc` (en récursif) à la commande `cpio` qui en effectue la sauvegarde. Ne surtout pas oublier le signe `>` lors de la sauvegarde.

Table 3. Options principales de la commande cpio

| Options | Description |
|---------|---|
| -o | Crée une sauvegarde (output). |
| -v | Affiche le nom des fichiers traités. |
| -F | Désigne la sauvegarde à modifier (support). |

Sauvegarde vers un support :

```
[root]# find /etc | cpio -ov > /dev/rmt0
```

Le support peut être de plusieurs types :

- `/dev/rmt0` : lecteur de bande ;
- `/dev/sda5` : une partition.

3.2. Type de sauvegarde

- Sauvegarde avec chemin relatif

```
[root]# cd /
[root]# find etc | cpio -o > /sauvegardes/etc.cpio
```

- Sauvegarde avec chemin absolu

```
[root]# find /etc | cpio -o > /sauvegardes/etc.A.cpio
```



Si le chemin indiqué au niveau de la commande "find" est en **absolu** alors la sauvegarde sera réalisée en absolu.

Si le chemin indiqué au niveau de la commande “find” est en **relatif** alors la sauvegarde sera réalisée en relatif.

3.3. Ajouter à une sauvegarde

Syntaxe de la commande cpio pour ajouter un contenu.

```
[cde de fichiers |] cpio {-o| --create} -A [-options] [<fic-liste] {F|>support}
```

Exemple :

```
[root]# find /etc/shadow | cpio -o -AF FicSyst.A.cpio
```

L'ajout de fichiers n'est possible que sur un support à accès direct.

| Option | Description |
|--------|--|
| -A | Ajoute un ou plusieurs fichiers à une sauvegarde sur disque. |
| -F | Désigne la sauvegarde à modifier. |

3.4. Compresser une sauvegarde

- Sauvegarder **puis** compresser

```
[root]# find /etc | cpio -o > etc.A.cpio
[root]# gzip /sauvegardes/etc.A.cpio
[root]# ls /sauvegardes/etc.A.cpio*
/sauvegardes/etc.A.cpio.gz
```

- Sauvegarder **et** compresser

```
[root]# find /etc | cpio -o | gzip > /sauvegardes/etc.A.cpio.gz
```

Il n'y a aucune option, comme pour la commande tar, permettant de sauvegarder et de compresser en même temps. Cela s'effectue donc en deux temps : la sauvegarde puis la compression.

La syntaxe de la première méthode est plus facile à comprendre et à retenir, car elle s'effectue en deux temps.

Pour la première méthode, le fichier de sauvegarde est automatiquement renommé par l'utilitaire gzip qui rajoute .gz à la fin du nom de ce fichier. De même l'utilitaire bzip2 rajoute automatiquement .bz2.

3.5. Lire le contenu d'une sauvegarde

Syntaxe de la commande cpio pour lire le contenu d'une sauvegarde cpio.

```
cpio -t [-options] [<fic-liste]
```

Exemple :

```
[root]# cpio -tv </sauvegardes/etc.152.cpio | less
```

| Options | Description |
|---------|-------------------------------------|
| -t | Lit une sauvegarde. |
| -v | Affiche les attributs des fichiers. |

Après avoir réalisé une sauvegarde, il faut lire son contenu pour être certain qu'il n'y a pas eu d'erreur.

De la même façon, avant d'effectuer une restauration, il faut lire le contenu de la sauvegarde qui va être utilisée.

3.6. Restaurer une sauvegarde

Syntaxe de la commande cpio pour restaurer une sauvegarde.

```
cpio {-i| --extract} [-E fichier] [-options] [<support]
```

Exemple :

```
[root]#cpio -iv </sauvegardes/etc.152.cpio | less
```

| Options | Description |
|---------|--|
| -i | Restauration complète d'une sauvegarde . |

| Options | Description |
|-------------------------|---|
| -E fichier | Restaure uniquement les fichiers dont le nom est contenu dans fichier. |
| -d | Reconstruit l'arborescence manquante. |
| -u | Remplace tous les fichiers même s'ils existent. |
| --no-absolute-filenames | Permet de restaurer une archive effectuée en mode absolu de manière relative. |



Par défaut, au moment de la restauration, les fichiers sur le disque dont la date de dernière modification est plus récente ou égale à la date de la sauvegarde ne sont pas restaurés (afin d'éviter d'écraser des informations récentes par des informations plus anciennes).

L'option -u permet au contraire de restaurer d'anciennes versions des fichiers.

Exemples :

- Restauration en absolu d'une sauvegarde absolue :

```
[root]# cpio -iv <home.A.cpio
```

- Restauration en absolu sur une arborescence existante :

```
[root]# cpio -iuv <home.A.cpio
```

L'option "u" permet d'écraser des fichiers existants à l'endroit où s'effectue la restauration. * Restauration en relatif d'une sauvegarde absolue :

```
[root]# cpio -iv --no-absolute-filenames <home.A.cpio
```

L'option longue "--no-absolute-filenames" permet une restauration en mode relatif. En effet le "/" en début de chemin est enlevé.

- Restauration en relatif d'une sauvegarde relative :

```
[root]# cpio -iv <etc.cpio
```

- Restauration en absolu du fichier « passwd » :

```
echo "/etc/passwd" > tmp;cpio -iuE tmp <etc.A.cpio; rm -f tmp
```

4. Utilitaires de compression - décompression

Le fait d'utiliser la compression au moment d'une sauvegarde peut présenter un certain nombre d'inconvénients :

- Allonge le temps de la sauvegarde ainsi que celui de la restauration.
- Rend impossible l'ajout de fichiers à cette sauvegarde.



Il vaut donc mieux effectuer une sauvegarde et la compresser qu'effectuer la compression lors de la sauvegarde.

4.1. Compresser avec gzip

Syntaxe de la commande gzip.

```
gzip [options] [fichier ...]
```

Exemple :

```
[root]# gzip usr.tar
[root]# ls
usr.tar.gz
```

Le fichier reçoit l'extension .gz.

Il conserve les mêmes droits et les mêmes dates de dernier accès et de modification.

4.2. Compresser avec bunzip2

Syntaxe de la commande bzip2.

```
bzip2 [options] [fichier ...]
```

Exemple :

```
[root]# bzip2 usr.cpio
[root]# ls
```

```
usr.cpio.bz2
```

Le nom du fichier reçoit l'extension .bz2.

La compression par “bzip2” est meilleure que celle par “gzip” mais dure plus longtemps.

4.3. Décompresser avec gunzip

Syntaxe de la commande gunzip.

```
gunzip [options] [fichier ...]
```

Exemple :

```
[root]# gunzip usr.tar.gz
[root]# ls
usr.tar
```

Le nom du fichier est tronqué par gunzip et se voit enlever l'extension .gz .

Gunzip décompresse également les fichiers portant les extensions suivantes :

- .Z ;
- -Z ;
- _Z.

4.4. Décompresser avec bunzip2

Syntaxe de la commande bzip2.

```
bzip2 [options] [fichier ...]
```

Exemple :

```
[root]# bunzip2 usr.cpio.bz2
[root]# ls
usr.cpio
```

Le nom du fichier est tronqué par bunzip2 et se voit enlever l'extension .bz2 .

bunzip2 décompresse également le fichier portant les extensions suivantes :

- -bz ;
- .tbz2 ;
- tbz.

