Administration Linux
Sauvegarde

La sauvegarde

La sauvegarde est un travail important de l'administrateur puisqu'en cas de gros problème, on passe généralement par une restauration du système depuis une sauvegarde, ou une image du système lorsque celui-ci était encore intègre (bon fonctionnement, pas de corruption). Chaque Unix est fourni avec des commandes et des procédures de sauvegarde qui lui sont propres. On distingue tout de même quelques outils communs.

Commandes, Plan, Scripts

- Pour la sauvegarde de fichiers et d'arborescences, utilisez les commandes tar et cpio. Ces commandes sauvent une arborescence, et un système de fichiers. On peut faire coïncider les deux.
- Pour la sauvegarde physique de disques et de systèmes de fichiers (des dumps), utilisez la commande dd.

Commandes, Plan, Scripts

Une sauvegarde **incrémentale** consiste à sauvegarder une première fois la totalité des données, puis **ensuite uniquement les fichiers modifiés**. L'administrateur aura parfois à définir des scripts de sauvegarde et de restauration adaptés au cas par cas (partition système, données applicatives...) et à automatiser quand c'est possible l'exécution de ceux-ci en fonction de la date, l'heure ou la charge de la machine

Commandes, Plan, Scripts

Il sera aussi très important de définir un plan de sauvegarde, en se posant les bonnes questions:

- Que faut-il sauvegarder ?
- Avec quelle fréquence ?
- Combien de temps conservera-t-on les sauvegardes, à quel endroit, en combien d'exemplaires ?
- À quel endroit sera stocké l'historique des sauvegardes ?
- Quel est le support le plus approprié ?
- Quels sont les besoins, en capacité, du support de sauvegarde ?
- Combien de temps prévoit-on pour sauvegarder un fichier, un système de fichiers et est-ce raisonnable ?
- La sauvegarde doit-elle être automatique ou manuelle ?
- Quelle est la méthode de sauvegarde la plus appropriée ?

La commande Tar

La commande **tar** est simple et efficace.

Elle crée des archives des fichiers, y compris l'arborescence de fichiers, dans une autre fichier (archive à l'extension .tar).

L'archive contient tous les fichiers et répertoires, mais aussi les uid, gid, permissions et dates de chaque fichier et répertoire archivé. Ceci permettra de restaurer tout ou partie de cette archive en cas de besoin, avec les propriétés et permissions correctes.

L'archive ainsi créée peut s'étendre sur plusieurs volumes c'est à l'utilisateur d'en insérer une nouvelle et la sauvegarde/restitution continue.

La commande Tar

La syntaxe de la commande tar est la suivante :

tar options noms

Une **option** est une lettre précédée d'un signe – et suivie d'éventuels paramètres.

On peut combiner plusieurs options (–x –v –z peut s'écrire –xvz).

Les autres arguments de la commande sont des noms de fichiers ou de

répertoires.

Un nom de répertoire désigne ce répertoire et tous les fichiers et les sous–répertoires qu'il contient récursivement.

La commande Tar - Archiver

La syntaxe est la suivante :

tar cvf nom_archive Fichier(s)

Exemple:

tar -cvf desktop.tar Desktop/

Les paramètres sont les suivants :

- **c** : création d'archive,
- **v** : mode bavard, tar indique ce qu'il fait,(verbose)
- **f** : le paramètre suivant est le nom de l'archive

La commande Tar - Lister

La syntaxe est :

tar tvf nom_archive

Exemple

Pour lister de l’archive précédente:

tar tvf desktop.tar

Le paramètre **t** liste le contenu de l’archive.

La commande Tar - Restauration

Pour restaurer le contenu d'une archive la syntaxe est :

tar xvf nom_archive fichiers

Pour restaurer l'archive précédente :

tar xvf desktop.tar

Le paramètre **x** permet l'extraction de l'ensemble des fichiers de l'archive, ou du ou des fichiers spécifiés à la suite du nom de l'archive.

La commande Tar - Autres paramètres

La commande tar permet de gérer les formats de compression directement :

- **z** : l'archive est compressée au format **gzip**.
- **Z** : l'archive est compressée au format **compress**.
- **j** : l'archive est compressée au format **bzip2**.

La commande Tar - Format gzip

Ainsi les commandes précédentes pour le format de compression **gzip** deviennent:

Pour la création d'archive compressée:

tar cvzf desktop.tar.gz Desktop/

Pour la restauration:

tar xvzf desktop.tar.gz Desktop/

La commande cpio

La commande **cpio** sauvegarde sur la sortie standard les fichiers dont on saisit les noms sur l'entrée standard, par défaut l'écran et le clavier. Il faut donc utiliser les redirections.

Cpio ne compresse pas les archives. C'est à vous de le faire.

La commande cpio - Archiver

La syntaxe générale est :

cpio –oL

Les paramètres les plus utilisés sont :

- -o : output, création de la sauvegarde en sortie.
- -L : sauve les fichiers liés et pas les liens symboliques.
- -v : mode bavard « verbose ».
- -c : sauvegarde des attributs des fichiers sous forme ASCII (pour l’échange entre divers OS).

La commande cpio - Archiver

Voici comment archiver et compresser le répertoire Desktop :

find Desktop -print| cpio -ocv | gzip > archive.cpio.gz

La commande cpio - Lister

La syntaxe générale est :

cpio -it archive

Les paramètres sont :

- -i : lecture de l’archive en entrée.
- -t : comme pour tar, liste le contenu de l’archive.

La commande cpio - Restaurer

La syntaxe générale est :

cpio -i[umd]

- -u : restauration inconditionnelle, avec écrasement des fichiers qui existent déjà. Par défaut les fichiers ne sont pas restaurés si ceux présents sur le disque sont plus récents ou du même âge.
- -m : les fichiers restaurés conservent leur dernière date de modification.
- -d : cpio reconstruit l'arborescence des répertoires et sous-répertoires manquants.

La commande cpio - Restaurer

Pour restaurer l’archive précédente :

cat archive.cpio.gz | gzip -cd | cpio –iuvd

La commande dd

La commande **dd** (device to device) est destinée à la copie physique, bloc par bloc, d'un fichier périphérique vers un fichier périphérique ou quelconque.

À l'origine elle était utilisée pour la lecture et l'écriture sur bande magnétique, mais elle peut être employée avec n'importe quel fichier. La commande **dd** permet de réaliser des copies physiques de disques et de systèmes de fichiers.

La commande dd

Argument	Rôle
if=fichier	Nom du fichier en entrée (celui à copier).
of=fichier	Nom du fichier en sortie
bs=n	Taille du bloc en octets.
count=n	Nombre de blocs à copier.
skip=n	Nombre de bloc à sauter au début du fichier d'entrée.
conv=	Conversion de l'entrée.
seek=	Nombre de blocs à sauter au début du fichier de sortie.
-s	Shell (commande de connexion) par défaut de l'utilisateur (variable SHELL). L'utilisateur peut le changer via la commande chsh.
-p	Le mot de passe de l'utilisateur. Attention ! le mot de passe doit déjà être crypté ! Aussi à moins de recopier le mot de passe d'un compte générique, vous préférerez utiliser ensuite la commande passwd.

La commande dd

L’option **conv** admet les paramètres suivants :

- ascii : convertir l’EBCDIC en ASCII.
- ebcdic : convertir l’ASCII en EBCDIC.
- block : compléter les blocs se terminant par un saut de ligne avec des espaces, jusqu’à atteindre la taille mentionnée par bs.
- unblock : remplacer les espaces en fin de blocs (de taille cbs) par un saut de ligne.
- lcase : transformer les majuscules en minuscules.
- ucase : transformer les minuscules en majuscules.
- noerror : continuer même après des erreurs de lecture.
- notrunc : ne pas limiter la taille du fichier de sortie.
- sync : compléter chaque bloc lu avec des NULs pour atteindre la taille ibs.

Administration Linux
L’horloge

La commande date

Pour connaître l'heure, utilisez la commande **date**.

Elle permet de donner la date actuelle, mais aussi de calculer d'autres dates en fonction soit de la date actuelle, soit en fonction d'une date quelconque. Date permet aussi de modifier la date et l'heure du système.

Par défaut la date affichée est la date (et l'heure) locale, configurée en fonction du fuseau horaire. Pour afficher l'heure UTC:

date --utc

La commande date

Le format de la date peut être modifié

Dans ce cas la syntaxe est:

date +"format"

La commande date

Voici quelques exemples de format possible :

Format	Résultat
% H	L'heure au format 00..23.
% M	Minutes 00..59.
% S	Secondes 00..60.
% T	Heure actuelle sur 24 heures.
% r	Heure actuelle sur 12 heures.
% Z	Fuseau horaire.
% a	Jour abrégé (lun, mar, etc.).
% A	Jour complet.
% b	Mois abrégé.
% B	Mois complet

La commande date

Format	Résultat
% d	Jour du mois.
% j	Jour de l’année.
% m	Numéro du mois.
% U	Numéro de la semaine 00..53.
% y	Deux derniers chiffres de l’année.
% Y	Année complète

La commande date

Pour afficher la date complète:

date +"Nous sommes le %A %d %B %Y, il est %H heures, %M minutes et %S secondes"

Les mots clés **today**, **yesterday**, **tomorrow**, **day(s), week(s), month(es), year(s), hour(s), minute(s), second(s)** sont acceptés, avec **+** (ajout à la date) ou **-** ou **ago** (retranche à la date précisée). Si la date n'est pas précisée, c'est la date en cours.

La commande hwclock

La commande **hwclock** permet d'interroger directement l'horloge matérielle RTC.

Le paramètre **--show** (par défaut) affiche la date actuelle.

Elle est différente du temps système provenant de **ntp** ou **date**. La fin de l'affichage donne d'ailleurs le décalage.

Exemple

hwclock --show

NTP

NTP (Network Time Protocol) est un protocole qui permet de synchroniser les horloges des ordinateurs via le réseau, notamment TCP/IP et donc Internet.

Les horloges des ordinateurs peuvent parfois avancer ou retarder.

Il y a de nombreux domaines où il est inadmissible de ne pas avoir un système à l'heure notamment pour des raisons de synchronisation très précises

NTP

Un serveur **NTP** diffuse l'heure au format UTC. Le client récupère l'heure et l'adapte en fonction de son fuseau horaire. Le serveur ne gère pas non plus les changements d'heure.

Pour peu que le serveur NTP soit à jour, l'heure est très précise. Elle est codée sur 64 bits :

- les 32 premiers bits donnent le nombre de secondes depuis le 1 e r

janvier 1900 à minuit (donc le bug NTP aura lieu avant le bug Unix)

- les 32 derniers bits donnent la précision des secondes.

Le nouveau protocole NTP4 donne une précision des secondes sur 64 bits, évitant ainsi un bug gênant dans lefutur

Client ntp

Le service ntpd permet de synchroniser une machine auprès d'un serveur de temps.

Le fichier de configuration est /etc/ntpd.conf**.** En principe ce fichier contient déjà un certain nombre de lignes qu'il faut éviter de toucher. Vous pouvez, voire devez, rajouter une ligne pointant sur le serveur de temps que vous avez choisi et relancer le service après.

Client ntp

Vous pouvez forcer une synchronisation manuelle avec la commande ntpdate. Celle-ci prend pour paramètre un nom de serveur ntp

ntpdate nomserveurntp

vous pouvez placer la commande de synchronisation en crontab tous les jours, ou toutes les heures.