

IV. Archivage - Planification

Sommaire

1. Archivage et compression
2. Date sous Linux
3. AT : planification ponctuelle
4. Cron : planification répétitive

Archivage & compression

Définition :

Compression: Réduire la taille d'un fichier par un algorithme de compression

Archivage : Placer un ensemble de fichiers et /ou dossiers dans un seul fichier

Compression sans archivage : gzip; gunzip; bzip2

Cependant ces outils ne permet de compresser qu'un seul fichier à la fois

solution: Archivage avec ou sans compression : tar ; star

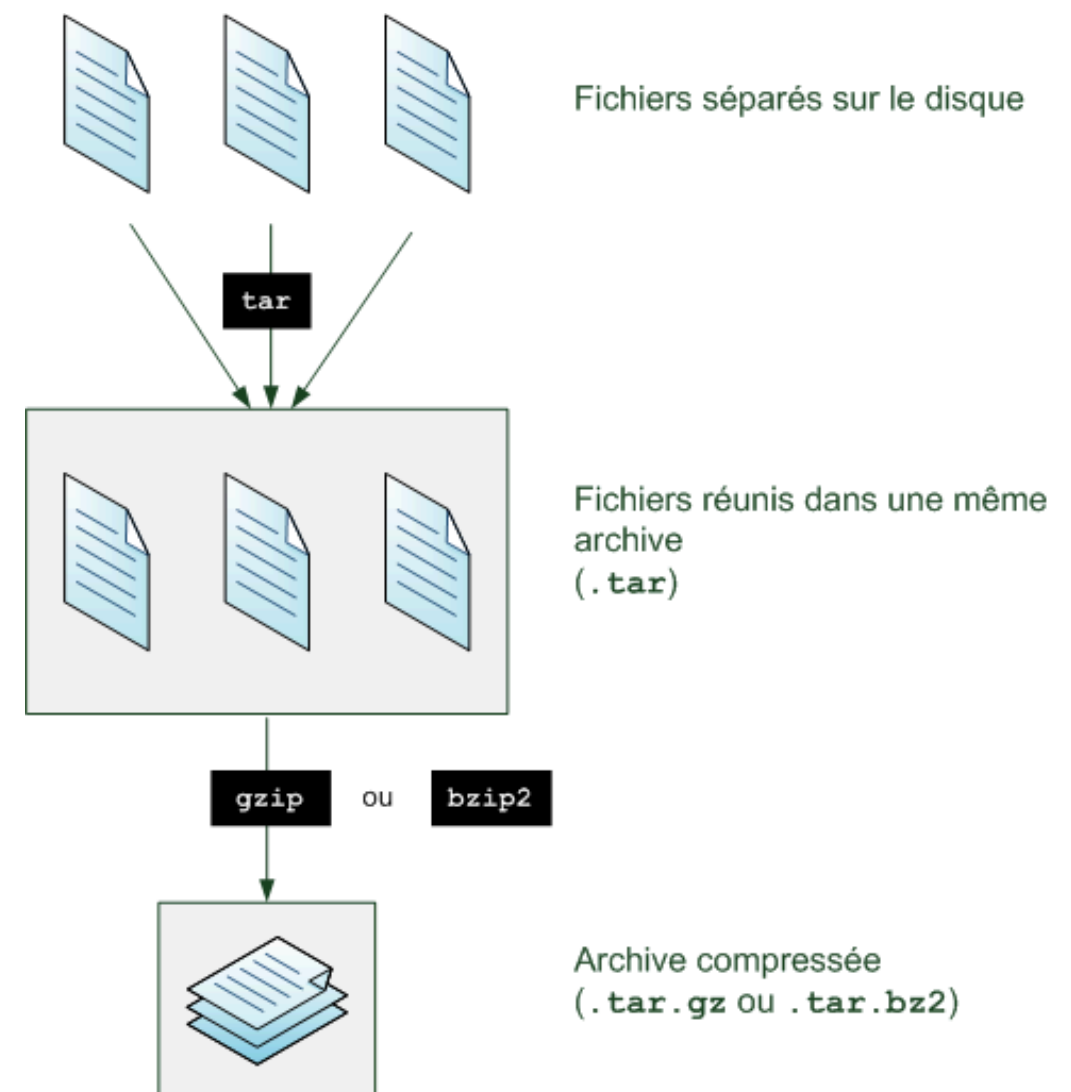
Archivage & compression

Sous Linux, on a depuis longtemps pris l'habitude de procéder en deux étapes :

1- réunir les fichiers dans un seul gros fichier appelé **archive**. On utilise pour cela le programme **tar** ;

2- compresser le gros fichier ainsi obtenu à l'aide de **gzip** ou de **bzip2**.

Ces deux étapes sont résumées dans le schéma **de la figure suivante**.



Archivage & compression

Archivage : Utilisation de la commande tar

Etape 1 : Regrouper les fichiers dans un dossier

```
$ls
Examples  projet.txt  Prive
licence1.txt  licence2.txt  Docs
exo.txt      Public
```

Par exemple, regroupons les fichiers d'extension txt dans le dossier Docs.

```
$mv *.txt Docs/
```

Le dossier Docs contient maintenant les fichiers .txt

Archivage & compression

Archivage : Utilisation de la commande tar

Etape 2 : Création de l'archive

Pour créer l'archive on utilise la commande tar avec ces options

-c : signifie crée l'archive

-v : signifie afficher les détails de l'opération

-f : signifie assembler l'archive dans un fichier

Syntaxe : tar -cvf fichier_archive.tar. dossier_a_archiver/

```
$ tar -cvf docs.tar Docs/
```

les options **-tf** : permet **d'afficher** le contenu de l'archive

```
$ tar -tf docs.tar  
Docs/  
Docs/exos.txt  
Docs/licence1.txt  
Docs/licence2.txt
```

Archivage & compression

Archivage : Utilisation de la commande tar

Ajouter un fichier supplémentaire à l'archive avec l'option -rvf

```
$ tar -rvf docs.tar examen.txt
```

les options **-xvf** : permet **d'extraire** les fichiers de l'archive

```
$ tar -xvf docs.tar  
Docs/  
Docs/exos.txt  
Docs/licence1.txt  
Docs/licence2.txt  
examen.txt
```

Les fichiers s'extraient dans le répertoire dans lequel vous vous trouvez. Vérifiez donc avant de les extraire que ceux-ci sont réunis dans un même dossier (avec -tf) si vous ne voulez pas que ces fichiers aillent se mélanger à d'autres !

Archivage & compression

Compression : Utilisation de la commande gzip et bzip2

Ces programmes sont simples à utiliser. Ils prennent comme paramètre le nom du fichier à compresser. Ils le compressent et modifient ensuite son nom.

Concrètement, ils ajoutent un suffixe pour indiquer que l'archive a été compressée :

.tar.gz : si l'archive a été compressée avec **gzip** ;

.tar.bz2 : si l'archive a été compressée avec **bzip2**.

```
$ gzip docs.tar
```

Pour décompresser

```
$ gunzip docs.tar.gz
```

```
$ bzip2 docs.tar
```

Pour décompresser

```
$ bunzip2 docs.tar.bz2
```


Archivage & compression

Archiver et Compresser : Utilisation de la commande tar

Durant l'archivage la commande tar est capable d'appeler la commande compression :

Les options `-zcvf` : permet d'archiver et de compresser en `gzip`

```
$ tar -zcvf docs.tar.gz Docs/
```

Les options `-jcvf` : permet d'archiver et de compresser en `bzip2`

```
$ tar -jcvf docs.tar.bz2 Docs/
```

Pour décompresser, c'est pareil, sauf que le `-c` est remplacé par un `-x` comme tout à l'heure :

```
$ tar -zxvf docs.tar.gz  
Ou  
$ tar -jxvf docs.tar.bz2
```

Planification

Sous Linux, il est possible d'automatiser des taches c'est à dire :

- Planifier l'exécution différée d'une action. Pour cela, on utilise le démon AT qui permet de configurer des tâches qui seront exécutées une seule fois à une date précise. Ex : exécuter une opération à aujourd'hui 14h15
- Planifier l'exécution périodique d'une action. Pour cela, on utilise le démon CRONTAB qui permet de configurer des taches qui sont exécutées par CRON. Ainsi on pourra exécuter une opération tous les jours, toutes les 15 minutes,

Commande Date

La commande **date** permet d'afficher la date , l'heure courant et le fuseau horaire
\$date

mercredi 3 avril 2019, 09:27:25 (UTC)

Les formats d'affichage

%H : heure

%M : minutes

%S : seconde

%Y : année

%m : mois

%d : jour

\$date "+%H:%M:%S"

09:40:45

\$date "+%Hh-%Mmn-%Ss"

09h-40mn-45s

\$date "+ %d-%m-%Y à %Hh:%Ms:%Ss "

03-04-2019 à 11h:08s:19s

Commande Date

Pour configurer la date système , on utilise l'option -s ou --set

```
$date -s "04/10/2019 13:24:45"
```

```
mercredi 10 avril 2019, 13:24:45 (UTC+0000)
```

```
mercredi 3 avril 2019, 09:27:25 (UTC)
```

Pour reconfigurer le timezone

```
$dpkg-reconfigure tzdata
```

```
Current default time zone: 'Africa/Dakar'
```

```
Local time is now:    Fri Apr  5 11:32:01 GMT 2019.
```

```
Universal Time is now: Fri Apr  5 11:32:01 UTC 2019.
```

Pour configurer la date automatique via un serveur de temps

Installer le paquet ntp

```
$apt-get install ntp
```

Démon Crontab

Cron est un logiciel (plus précisément un démon) intégré dans tout système Unix qui permet d'exécuter une commande à intervalles régulières avec un réglage extrêmement précis.

Par exemple, vous pouvez sauvegarder vos fichiers de configuration tous les 2 jours à 13h37 sauf le dimanche et le lundi (c'est un exemple comme un autre).

Cron est donc un programme qui tourne en arrière-plan en attendant qu'un évènement spécifié dans son fichier de configuration (crontab) se réalise.

Démon Crontab

Le fichier de configuration : /etc/crontab

Les commandes de bases :

- Afin de lister le contenu de la crontab :

Crontab -l

- Pour lister la crontab d'un utilisateur on ajoute l'option -u : **Crontab -u -l**
- Pour éditer la crontab : **Crontab -e**

Remarque: A la première utilisation de crontab -e , il sera demandé l'éditeur à utiliser

Démon Crontab

Contrôle d'accès

- `cron.allow`: seuls utilisateurs habilités à programmer des tâches
- `cron.deny`: seuls utilisateur NON autorisés à programmer des tâches (suppose l'absence de `cron.allow`)
- si ni `cron.allow` ni `cron.deny`: seul root y a droit

Démon crontab

Le format d'une ligne crontab est bien précise, cette dernière étant indiquée dans le fichier crontab :

m h dom mon dow user command

25 6 * * * root cp /home/root/tp/projet.tar.gz /backup

Colonne	Description	Valeur
m	Minute	0 à 59
h	Heure	0 à 23
dom	Jour du mois (Day of Month)	1 à 31
mon	Mois	1 à 12
dow	Jour de la semaine (Day of Week)	0 à 6 (0=Dimanche, 6=Samedi)
user	Utilisateur qui va exécuter la commande	
command	La commande à exécuter	

Démon crontab

Il est possible d'utiliser les caractères suivants :

- ***** : toutes les valeurs de la plage autorisée
- **n** : un nombre ou n est dans la plage autorisée.
- **2,4,6** : différentes valeurs .
- **10-20** : plage de valeurs comprises dans la plage autorisée.
- ***/2** : valeurs multiples de 2.

Exemple :

m h dom mon dow user command

#A 20h le lundi et le vendredi, toute l'année :

0 20 * * 1,5 command

#A 21h30, du lundi au vendredi, toute l'année :

30 21 * * 1-5 command

#Toutes les 6 heures, tous les jours, toute l'année :

0 */6 * * * command

#Les dix premiers jours de chaque mois à 19h30 :

30 19 1-10 * * command

#Tous les jours pairs du mois à 23h30 :

30 23 */2 * * command

Démon AT

La commande **at** permet d'effectuer une action particulière dans la journée (par exemple dans 15 minutes ou à une heure précise), vous n'utiliserez pas cron (car il n'y a pas de répétition, c'est juste une exécution unique programmée)

Les unités de temps sont : minutes, hours, days, weeks, months or years

Pour utiliser la commande at :

Dans une console, tapez at *** (***) étant le temps souhaité, voir la liste plus bas) puis entrée.

Vous obtenez l'invite de commande at> ,

tapez la commande que vous souhaitez exécuter puis entrée.

Vous pouvez désormais taper une deuxième commande qui sera exécuté ou "Ctrl+D" afin de signifier que la série de commandes est terminée.

Démon AT

Exemple :

Je souhaite supprimer tous les fichiers *.tmp du répertoire tmp :

- dans 2 minutes :

 - at** now + 2 minutes

 - rm /tmp/*.tmp

 - Entrée "Ctrl+D"

- à 13h37 :

 - at** 13:37

 - rm /tmp/*.tmp

 - Entrée "Ctrl+D"

Démon AT

Exemple :

\$ at 16:45

\$ at 16:45 040619 #(Format MMDDYY)

\$ at midnight

\$ at now # très utile pour les tests

\$ at now + 2 hours # très utile pour les tests

\$ at 16:45 + 3 days

FIN COURS 4