

L'administration d'un serveur Apache sous Linux

Module 5 - L'administration Linux



Objectifs

- Savoir gérer les droits
- Savoir archiver et compresser
- Savoir installer des logiciels

L'administration Linux

Droits d'accès

- La configuration des droits d'accès est **importante** sous Linux
- Chaque fichier a un **propriétaire** et un **groupe d'appartenance**
- Rappel : la commande `ls -l` permet d'avoir les informations complètes :

```
-rw-r--r-- 1 root root 19605 oct. 25 2014 /etc/services
```

L'administration Linux

Masque

- Lors de la création d'un fichier, des droits de base sont posés
- Ces droits sont appliqués à partir d'un masque de droit
- Le masque se calcule par soustraction du droit maximal auquel on enlève le droit voulu et on obtient le masque :

Droit maximal fichier :	666
Droit voulu :	640
=====	
Masque	026

Masque

- La commande `umask` permet d'afficher le masque en cours :

```
$ umask  
0022
```

- Et de changer de façon interactive le masque :

```
$ umask 0026
```

- Attention, ce masque est identique pour les fichiers et les répertoires

Interprétation des droits

Droit d'accès	Fichier	Répertoire
r	Lecture	Visibilité du contenu du répertoire Nécessaire pour <code>ls</code>
w	Écriture	Possibilité de créer ou supprimer des fichiers ou des répertoires dans le répertoire concerné
x	Exécution	Possibilité de se positionner dans le répertoire avec <code>cd</code>

Changer l'utilisateur

- Syntaxe : **chown** <user>:<group> <fichier>
 - <user> : nom du nouveau propriétaire
 - <group> : nom du nouveau groupe
- Par défaut, seul root a tous les pouvoirs avec **chown**
- Exemples :

```
# chown user1:group1 fichier.txt
# chown user1 fichier2.txt
# chown -R :group2 dossier1
```

Gestion des droits

- Syntaxe : `chmod [options] <droits> <fichier>`
- Option intéressante : `-R` pour appliquer les droits de façon récursive
- Il existe deux méthodes pour appliquer les droits :
 - La [méthode octale](#)
 - La [méthode symbolique](#)

L'administration Linux

Méthode octale

- Avec cette méthode, chaque droit a une valeur, il faut les additionner pour obtenir le droit à mettre en place

Propriétaire			Groupe			Autres		
R	W	X	R	W	X	R	W	X
400	200	100	40	20	10	4	2	1

- Exemples :
 # `chmod 640 fichier.txt`
 # `chmod 755 dossier2`

Méthode symbolique

- On utilise :
 - les **symboles de droits** : r, w, x
 - les **symboles de propriétaires** : u, g, o (*others*), a (*all*)
 - les **opérateurs** : +, -, =
- Les symboles peuvent être combinés
- Exemples :

```
# chmod u-x,g-w,o+r fichier.txt  
# chmod a=rw fichier2.txt  
# chmod ug=rw,o=r fichier3.txt
```

L'administration Linux

Gérer les droits d'accès

Démonstration



L'administration Linux

Archivage

- L'archivage est la **concaténation** d'un ensemble de fichiers et répertoires en un **seul fichier**
- L'archivage permet de gérer plus facilement un envoi de plusieurs fichiers, par mail, par SCP, etc.
- La taille globale de l'archive est à peu près la somme des tailles des fichiers
- La commande standard est **tar**, la commande **cpio** existe aussi

L'administration Linux

Compression

- L'archivage va souvent être utilisé avec la compression pour réduire la taille de l'archive
- L'algorithme principalement utilisé est **gzip**
- L'algorithme **bzip2** est aussi de plus en plus utilisé
- Le taux de compression de bzip2 est plus efficace que celui de gzip, mais bzip2 est plus lent

L'administration Linux

Archiver

- Syntaxe : `tar [options] <nom archive> <fichiers à archiver>`
- Créer une archive compressée en gzip :

```
# tar cvzf monarchive.tar.gz /etc/ /root/
```

 - `-c` : créer l'archive
 - `-v` : mode verbeux
 - `-z` : compression gzip
 - `-f <fichier.tar>` : utilisation du fichier nommé

L'administration Linux

Désarchiver

- On fait les deux actions en même temps, décompression et désarchivage

```
# tar xvzf monarchive.tar.gz -C /tmp
```

- **-x** : désarchivage
- **-C <répertoire>** : désigne le répertoire de destination ; si non précisé, désarchive dans le répertoire courant

Autres commandes

- Lister le contenu d'une archive :

```
# tar tvzf monarchie.tar.gz
```

- Archiver et compresser en deux commandes :

```
# tar cvf archive.tar /etc/ /root/
```

```
# gzip archive.tar
```

- Décompresser et désarchiver en deux commandes :

```
# gunzip archive.tar.gz
```

```
# tar xvf archive.tar -C /tmp
```


L'administration Linux

La commande tar

Démonstration



Installer des logiciels

- Il y a trois façons d'installer des logiciels sous Linux :
 - À partir des **sources** (complexe)
 - À partir d'un **paquet téléchargé** (pas de gestion de dépendances)
 - À partir des **dépôts** (solution conseillée)

L'administration Linux

Les dépôts

- Trois briques fonctionnelles :
 - **Les paquets** : fichiers d'archive spécifiques (.deb pour Debian)
Comprend le logiciel concerné, les scripts d'installation, les fichiers annexes nécessaires.
Liste aussi les paquets dont il est dépendant.
 - **Le dépôt** : serveur web stockant les paquets, dispose d'une base de données les référençant
 - **Le miroir** : copie d'un dépôt
- La gestion est sécurisée grâce à des **clés GPG**

Configuration client

- Fichier de configuration : `/etc/apt/sources.list`.
- Contient plusieurs lignes désignant les dépôts à utiliser
- Exemple :

```
deb http://ftp.fr.debian.org/debian/ jessie main  
deb-src http://ftp.fr.debian.org/debian/ jessie main
```
- `deb` : paquets binaires ; `deb-src` : paquets sources

Configuration client

- L'**URL du dépôt** peut être en HTTP, HTTPS, FTP, CDROM...
- Le champ suivant est la **version de la distribution** : Jessie (version 8) ou stable (dernière version 9), testing (prochaine stable)...
- Le dernier champ correspond aux **sections** à utiliser :
 - **main** : paquets Debian standards, sous licence libre
 - **contrib** : paquets libres ayant des dépendances non libres
 - **non-free** : paquets sous licence non libre

L'administration Linux

Configuration client

```
deb http://ftp.debian.org/debian/ jessie main contrib non-free
deb-src http://ftp.debian.org/debian/ jessie main contrib non-free
```

```
deb http://security.debian.org/ jessie/updates main contrib non-free
deb-src http://security.debian.org/ jessie/updates main contrib non-free
```

```
# jessie-updates, previously known as 'volatile'
deb http://ftp.debian.org/debian/ jessie-updates main contrib non-free
deb-src http://ftp.debian.org/debian/ jessie-updates main contrib non-free
```

```
# jessie-backports, previously on backports.debian.org
deb http://ftp.debian.org/debian/ jessie-backports main contrib non-free
deb-src http://ftp.debian.org/debian/ jessie-backports main contrib non-free
```

Gestion des paquets

- Pour gérer les paquets, on utilise les commandes **aptitude** ou **apt**
- Ces outils permettent de gérer les dépendances
- Ces commandes utilisent un **cache local** qu'il faut mettre à jour régulièrement
- Ce cache contient la liste des paquets et leur version

Gestion des paquets

- Mettre à jour le cache :
 - # `apt-get update`
 - # `aptitude update`
- Rechercher un paquet :
 - # `apt-cache search <expression>`
 - # `aptitude search <expression>`
- Mettre à jour l'ensemble des paquets installés :
 - # `apt-get upgrade`
 - # `aptitude upgrade`

Gestion des paquets

- Installer un paquet :

```
# apt-get install <paquet>
```

```
# aptitude install <paquet>
```

- Supprimer un paquet :

```
# apt-get remove - -purge <paquet>
```

```
# aptitude remove - -purge <paquet>
```

L'administration Linux

Le gestionnaire de paquets

Démonstration



Conclusion

- Vous savez maintenant gérer les droits
- Vous savez archiver et compresser des répertoires
- Vous savez installer des applications supplémentaires