Administration SI et outils reseaux sur linux

Table of Contents

1. LINUX CLI - Cheat sheet

1. Connaitre la version de linux.

```
uname ou lsb-release
```

```
uname -a | -v | -r
```

```
lsb-release -a
```

2. Linux commande CLI

Table 1. Commande text dans la CLI

Commande	Description
`CTRL + U`	couper le texte jusqu'au curseur
`CTRL + K`	couper le text depuis le curseur jsuqu'à la fin de la ligne
`CTRL + Y`	coller texte du presse papier
`CTRL + E`	deplacer curseur jusq'à la fin
`CTRL + A`	deplacer le curseur jusqu'à la fin de la ligne
`ALT + F:`	sauter au prochain espace
`ALT + B:`	revenir au dernier espace
`ALT + Backspace`	supprimer en arrière
`CTRL + W`	couper ce qui est derrière le curseur
`Shift + insert`	coller le text dans le terminale

Commande de base

- `apt` : il y a trois commandes que vous devez connaître pour utiliser pleinement APT: add-apt-repository(pour localiser des paquets tiers), apt-get(pour installer réellement des paquets), et apt-cache(pour rechercher vos dépôts). Si votre distribution n'utilise pas APT, elle peut utiliser YUM, RPM ou une autre alternative. Regardez dans leurs commandes équivalentes.
- `bg / fg` : Envoie un travail de premier plan à exécuter en arrière-plan ou un travail d'arrière-plan à exécuter en premier plan. Pour plus d'informations sur les travaux, voir lajobscommande.
- `df` : Affiche la quantité d'espace utilisée et libre sur votre système.
- `free` : Affiche la quantité de RAM utilisée et gratuite sur votre système.
- `ip` : Affiche des informations utiles sur le réseau telles que votre adresse IP, les interfaces réseau, l'utilisation de la bande passante, etc. Peut également être utilisé pour configurer les paramètres liés au réseau.

- `jobs` : Affiche tous les travaux en cours et leurs statuts. Un travail est juste une représentation d'un processus en cours ou d'un groupe de processus.
- `kill / killall` : Vous pouvez utiliserkillpour terminer un processus en fonction de son ID de processus (souvent utilisé conjointement avec lapscommande) alors que vous pouvez utiliserkillallpour terminer tous les processus dont les noms correspondent à votre requête.
- `mount / umount`: Attache et détache un système de fichiers distinct du système de fichiers principal de votre système. Principalement utilisé pour rendre les périphériques externes, tels que les disques durs ou les clés USB, interactifs avec votre ordinateur.
- `ps` : Affiche une liste des processus en cours d'exécution. Par défaut, il ne répertorie que les processus démarrés sous votre utilisateur actuel, mais des paramètres existent pour trouver et filtrer toutes sortes de processus.
- `sudo / gksudo` : La pré -programmationsudovous permet d'exécuter n'importe quelle commande en tant que super-utilisateur (par exemplesudo [command1]). Si vous souhaitez exécuter un programme graphique avec des privilèges super-utilisateur, utilisezgksudosuivi du fichier exécutable du programme.
- `top` : Affiche une liste des processus en cours d'exécution, triés en fonction du nombre de processeurs utilisés par chaque processus. Contrairement àpscette commande, cette commande est mise à jour régulièrement en temps réel. Fondamentalement, un terminal équivalent à Gestionnaire des tâches.
- `uname` : Affiche les informations du système de base en fonction des paramètres que vous utilisez, tels que le nom et la version du noyau, le matériel de la machine et le système d'exploitation.
- `uptime` : Affiche le temps écoulé depuis le dernier démarrage.
- `whereis` : Trouve l'emplacement du fichier exécutable pour un programme donné.
- `whoami` : Affiche le nom d'utilisateur actuel. C'est pratique lorsque vous passez d'un utilisateur à l'autre avec lasucommande et que vous perdez la trace de qui vous êtes en ce moment.

Commande navigation terminale

- `&&` : Celui-ci est si basique que ce n'est même pas une commande technique. Si vous souhaitez exécuter plusieurs commandes dans un ordre séquentiel, placez-les entre chaque commande. Par exemple, [command1] && [command2]exécutera d'abord [command1] puis le suivra immédiatement avec [command2]. Vous pouvez enchaîner autant de commandes que vous le souhaitez.
- `!` : Répète une commande récemment utilisée. Le mieux est de l'utiliser en conjonction avec la historycommande. Vous pouvez utiliser !npour répéter la commande n-ème dans l'histoire. Vous pouvez également utiliser !-npour répéter la commande qui est arrivée il y a des commandes.
- `cd` : Change le répertoire actuel du terminal.
- `clear` : Efface l'écran du terminal.
- `history` : Affiche une liste de toutes les commandes récemment utilisées. Vous pouvez également faire défiler les commandes récemment utilisées en appuyant sur les flèches haut et bas du

terminal.

- `ls` : Affiche une liste de tous les fichiers du répertoire actuel du terminal. Vous pouvez le modifier avec des paramètres pour spécifier un autre répertoire ou pour changer le format de la liste.
- `man` : Affiche une page d'aide (à partir du manuel) basée sur votre requête de recherche. Très utile pour apprendre à utiliser une commande que vous ne reconnaissez pas ou lorsque vous oubliez les paramètres d'une commande rarement utilisée. Si jamais vous êtes confus, tournez-vous vers l'homme.
- `pwd` : Affiche le répertoire du terminal actuel en tant que chemin absolu.
- `whatis` : Affiche une brève description des programmes en ligne de commande. Pensez-y comme une version simplifiée de manquand vous n'êtes pas sûr de ce qu'une commande fait, mais n'avez pas besoin du manuel complet sur la façon de l'utiliser.

Commande gestion de fichier

- `cat` : Lorsqu'il est utilisé sur un seul fichier texte, il affichera le contenu de ce fichier. Lorsqu'il est utilisé sur deux ou plusieurs fichiers texte, il affiche tous leurs contenus dans un ordre séquentiel. Utilisez l'opérateur de redirection (" > ") pour combiner plusieurs fichiers texte en un seul fichier texte.
- `chmod / chown` : Lachmodcommande modifie les permissions de lecture, d'écriture et d'exécution d'un fichier pendant que lachowncommande change l'utilisateur et / ou le groupe d'utilisateurs qui possède un fichier.
- `cp` : Fait une copie d'un fichier. Par défaut, la copie apparaît dans le répertoire du terminal actuel, mais vous pouvez également spécifier le répertoire de destination.
- `find` : Recherche un répertoire spécifique (ou l'intégralité de votre système) pour rechercher les fichiers correspondant à un ensemble de critères donné. Il existe des dizaines d'options, notamment le nom de fichier, le type de fichier, la taille du fichier, les permissions, les propriétaires, la date de création, la date de modification, etc.
- `grep` : Recherche un fichier ou un ensemble de fichiers spécifique pour voir si une chaîne de texte existe et, si c'est le cas, vous indique où le texte existe dans ces fichiers. Cette commande est extrêmement flexible (par exemple, utiliser des caractères génériques pour rechercher tous les fichiers d'un type donné) et particulièrement utile pour les programmeurs (pour trouver des lignes de code spécifiques).
- `locate` : recherche dans le système des fichiers ou des répertoires correspondant à la requête de recherche, puis affiche les chemins absolus pour chaque correspondance. Par défaut, il ne recherche que les répertoires pour lesquels vous avez des autorisations. C'est le moyen le plus simple et le plus rapide de trouver un fichier.
- `mkdir / rmdir` : Crée ou supprime un répertoire, par défaut dans le répertoire du terminal actuel, mais un répertoire cible peut également être spécifié. Lors de la suppression, le répertoire doit être complètement vide.
- `mv` : Déplace un fichier d'un répertoire à un autre et vous pouvez spécifier un nom différent pour le fichier dans le répertoire cible. Vous pouvez utiliser cette commande pour renommer un fichier en le déplaçant dans le même répertoire mais avec un nom de fichier différent.
- `nano / emacs / vim` : Les trois principaux éditeurs de texte de terminal qui existent sur presque

tous les systèmes Linux, classés par complexité croissante. Les débutants doivent s'en tenir aux nanodeux emacset vimsont extrêmement complexes (et extrêmement puissants).

- `Rename` : Modifie le nom d'un fichier ou d'un ensemble de fichiers. Livré avec beaucoup de paramètres intéressants, vous permettant de renommer automatiquement un tas de fichiers en fonction d'un modèle.
- `rm` : Supprime les fichiers. Avec un certain paramètre, il peut être utilisé pour effacer tout le contenu d'un répertoire spécifié. Il peut également être utilisé pour supprimer plusieurs fichiers qui correspondent tous à un certain modèle de nom de fichier.
- `touch` : Modifie la date d'accès ou la date de modification du fichier donné.
- `wget` : Télécharge le fichier ou la page à l'URL Web donnée.
- `zip / gzip / tar` : Divers formats pour compresser et décompresser les archives de fichiers.

2. Gestion de Disque

1. ls /mnt

> Lister les partions de disque.

2. less /etc/fstab

> Information sur la partition disque de linux.

3. df

> Connaitre l'espace disque.

3. Daemon linux

1. service (old) ou systemctl (new)

CLI: `df -f`

Systemd: Deamon linux gestionnaire du noyau linux.

CLI: `service [nomService] status | start | stop`

> connaitre le status du X service

CLI: `systemctl status | start | stop [nomService]`

> connaitre le status du X service avec la .CLIe systemctl

2. top et htop (monitoring process)

CLI: `systemctl status udev`

> top permet de visualiser la consomation de ressource des processus actives. htop est un outils améliorer de top.

4. Reseaux

1. netstat

Netstat ou Networking Statistique. Il permet de générer une présentation assez complete du reseau.

CLI: `netstat -nr`

```
Affiche la table de routage
```

CLI: `netstat -laputen | grep 80`

```
Connaitre le port 80
```

CLI: `netstat -tulpen`

```
Connaitre un max d'information sur les processus et avec sudo, on affiche les
processus impliqués
```

2. tcpdump

```
outils puissant qui permet d'afficher la trame TCP. On peut avoir à l'utiliser pour
analyser les failles réseaux, le 3 hands check (SYN,SYN[ACK],ACK)
```

3. nmap

nmap est l'accronyme de network map. nmap permet de générer une cartographie de l'adresse ip demandé et affiche les détails de port ouvert de l'hote.

4. hosts

CLI: `nmap [monIP ou monDomainName]`

`/etc/hosts` est le fichier de configuration de l'hote de la machine en question. Il permet de spécifier le DNS c'est à dire, de definier dans ce fichier /etc/hosts la correspondance entre un ip et un nom de domaine.

NOTE	*/etc/hosts* exemple: 127.0.0.1 localhost 192.168.1.10 patsou.ddns.net

NB: si je veux ajouter exemple xxx.com sur mon ip public, je fais:

1. aller sur `network-tools.com/`
2. recupérer un adresse ip
3. dans /etc/hosts/
4. je rajoute `adresseIP_X xxx.com`

5. ssh : authentification à distance.

Outil permettant d'accéder à une machine distante

CLI: `ssh-keygen`

Permet de générer une clé `ssh` facilement sur linux. la .CLIe `ssh-keygen -t rsa` est très utile.

Démarche:

1. ssh-keygen -t rsa
2. stocker la clé dans $HOME
3. voir la description du contenu: `cat $HOME/.ssh/id_rsa.pub`
4. copier la clé sur le serveur: `ssh-copy-id utilisateur@ipduserveur`
5. ssh 'user@ipserveur'

```
et c'est bon! :), vous êtes authenfifié en toute securité. :)
```

6. sftp

> Transfert de fichier sécurisé.

7. stat

> acronyme de statisitque de fichier ou du system. Il permet de voir les détails d'un fichier, création, droit, etc.

8. less | cat

> decrire un fichier.

5. processus

1. lsof

- la commande `lsof` remplace TOUT :)

> lister les fichiers d'un processus en cours d'execution.

```
`sudo lsof | head`
```

exemple:

on va voir un processus qui tourne et voir les fichiers qu'il a ouvert

> on va voir le processus

```
`sudo nestat -tulpen`
```

> on va ouvrir le log du programme

```
`lsof /etc/log/[nomProgramme].log`
```

1.1. ps VS lsof: binaire des processus?, où sont ils?

```
`lsof -p [PIDduProcess]| grep log`
```

```
`ps aux | grep yyy`
```

> voir les processus en cours d'execution.

1.2. ps VS lsof: connaitre les processus qu'utilise une librairie

```
`lsof [cheminDuLibrairie]`
```

> mieux que ps :).

exemple:

```
ls -alh /lib/i386-linux-gnu/libgcc_s.so.1
-rw-r--r-- 1 root root 114K avril  7  2017 /lib/i386-linux-gnu/libgcc_s.so.1
```

```
lsof /lib/i386-linux-gnu/libgcc_s.so.1
lsof: WARNING: can't stat() fuse.gvfsd-fuse file system /run/user/120/gvfs
```

sortie:

```
.CLI    PID   USER  FD   TYPE DEVICE SIZE/OFF    NODE NAME
TeamViewe 4603 patsoo mem    REG    8,3   116312 8659775 /lib/i386-linux-
gnu/libgcc_s.so.1
wineserve 5093 patsoo mem    REG    8,3   116312 8659775 /lib/i386-linux-
gnu/libgcc_s.so.1
services. 5137 patsoo mem    REG    8,3   116312 8659775 /lib/i386-linux-
gnu/libgcc_s.so.1
explorer. 5526 patsoo mem    REG    8,3   116312 8659775 /lib/i386-linux-
gnu/libgcc_s.so.1
TVGuiDele 5540 patsoo mem    REG    8,3   116312 8659775 /lib/i386-linux-
gnu/libgcc_s.so.1
```

1.3. Reseau avec lsof: connaitre les processus ecoutant un protocole

```
`lsof -i udp`
```

1.4. connaitre l'espace disque

> au lieu de faire `df -f`, on utilise `lsof`

```
`lsof -f`
```

1.5. lsof VS htop : monitoring un numéro de processus

```
`lsof -p [numPID]`
```

exemple:

1. htop
2. recupérer num PID
3. lsof -p [numPID]

   ```
   lsof va décrire tout ce qui se passe sur le PID (les fichiers ouverts, les ports,
   ...).
   ```

2. AWK

Le programme awk est une suite d'action de la forme : motif { action } , le motif permet de determiner sur quels enregistrements est appliquée l'action.

Un enregistrement est

une chaine de caractères séparée par un retour chariot, en général une ligne.

Un champs est

une chaine de caractères separée par un espace (ou par le caractère specifié par l'option -F), en générale un mot. On accède à chaque champs de l'enregistrement courant par la variable $1, $2, ... $NF. $0 correspond à l'enregistrement complet. La variable NF contient le nombre de champs de l'enregistrement courant, la variable $NF correspond donc au dernier champs.

Exemples:

```
`awk -F ":" '{ $2 = "" ; print $0 }'` /etc/passwd imprime chaque ligne du fichier
/etc/passwd après avoir effacé le deuxième champs
```

```
`awk 'END {print NR}'` fichier    imprime le nombre total de lignes du fichiers
```

```
`awk '{print $NF}'` fichier   imprime le dernier champs de chaque ligne
```

```
`who | awk '{print $1,$5}'`   imprime le login et le temps de connexion.
```

```
`awk 'length($0)>75 {print}'` fichier imprime les lignes de plus de 75 caractères.
(print équivaur à print $0)
```

Triches:

> lister les nombres d'appel des commandes linux

```
`history | awk '{print $2}' | sort | uniq -c | sort -rn | head -10``
```

> monitoring des LOGS en temps réel

```
sudo tail -f /var/log/apache2/access.log
sudo less +F  /var/log/apache2/access.log`
```