Commande Linux : I-Gestion des processus sous Linux

Un processus est un terme utilisé pour décrire une application ou un programme. Par exemple, lorsque nous ouvrons un navigateur Internet comme Google Chrome, nous pouvons dire qu'un processus, responsable de l'exécution de Chrome, a été lancé et est en cours d'exécution jusqu'à ce que nous fermions le navigateur. Même lorsque nous exécutons une commande bash, un nouveau processus apparaît. Si nous ouvrons la même application deux fois ou si vous et un ami l'ouvrez sur le même système, deux processus seront lancés.

Dans ce tutoriel, nous vous apprendrons à gérer les processus sous Linux via la ligne de commande. Cela est nécessaire si vous souhaitez voir les processus actifs ou si vous souhaitez voir les processus qui prennent la plupart des ressources de votre machine.

1. Visualisation des processus en cours

Voici des mots-clés utiles pour apprendre avant de commencer à lire ce didacticiel:

1- PID – ID de processus. Chaque processus possède son propre numéro. Il ne peut pas y avoir plus d'un même PID dans le système. Les processus sont référencés par un identifiant unique, le PID. Ce nombre peut être utilisé pour changer la priorité d'un processus ou pour l'arrêter.

2- PPID – ID de parent de processus. ID du processus qui a lancé ce processus en particulier. Un processus correspond à n'importe quel exécutable exécuté. Si le processus 2 a été lancé par le processus 1, on l'appelle un processus fils. Le processus qui l'a lancé est appelé processus parent.

```
bilel@bilel-G3-3590:~$ pstree
systemd─┬─ModemManager───2*[{ModemManager}]
        ├─NetworkManager───2*[{NetworkManager}]
        ├─accounts-daemon───2*[{accounts-daemon}]
        ├─acpid
        ├─agetty
        ├─atd
        ├─avahi-daemon───avahi-daemon
        ├─blkmapd
        ├─blueberry-tray─┬─python3───rfkill
        │                └─4*[{blueberry-tray}]
        ├─bluetoothd
        ├─colord───2*[{colord}]
        ├─containerd───18*[{containerd}]
        ├─cron
        ├─csd-printer───2*[{csd-printer}]
        ├─cups-browsed───2*[{cups-browsed}]
        ├─cupsd
        ├─dbus-daemon
        ├─dockerd───17*[{dockerd}]
        ├─gnome-keyring-d───3*[{gnome-keyring-d}]
        ├─irqbalance───{irqbalance}
        ├─2*[kerneloops]
        ├─lightdm─┬─Xorg───5*[{Xorg}]
        │         └─lightdm─┬─cinnamon-sessio─┬─agent───2*[{agent}]
        │                   │                 ├─applet.py
        │                   │                 ├─blueberry-obex-───3*[{blueberry-obex-}]
        │                   │                 ├─cinnamon-killer───3*[{cinnamon-killer}]
        │                   │                 └─cinnamon-launch───cinnamon─┬─MainThread─┬─Privileged Cont───+
        │                   │                                              │            ├─Web Content───35*[+
        │                   │                                              │            ├─Web Content───37*[+
        │                   │                                              │            ├─2*[Web Content───3+
        │                   │                                              │            ├─Web Content───23*[+
        │                   │                                              │            ├─WebExtensions───30+
        │                   │                                              │            └─69*[{MainThread}]
```

La commande **pstree** donne une bonne illustration de la hiérarchie des processus parents et fils.

Les options les plus courantes de **pstree** sont **-p** pour afficher les **PIDs** et **-h** pour faire ressortir (en gras) les processus utilisateurs.

Pstree -p

Pstree -h

Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000.

Commande Linux : I-Gestion des processus sous Linux

Les deux commandes les plus couramment utilisées pour visualiser les processus sont **top** et **ps**. La différence entre les deux est que top est utilisé de manière interactive/dans un terminal et que ps est plutôt utilisé dans les scripts, combiné avec d'autres commandes bash.

top – Cette commande est probablement la plus basique et est souvent utilisé pour afficher simplement les processus qui consomment le plus de ressources actuellement. Lorsque vous exécutez la commande top dans un terminal, vous verrez une fenêtre semblable à ceci:

```
top - 19:14:34 up 23:47,  1 user,  load average: 0.29, 0.31, 0.26
Tâches: 283 total,   1 en cours, 282 en veille,   0 arrêté,   0 zombie
%Cpu(s):  2.2 ut,  0.7 sy,  0.0 ni, 97.1 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  23883.5 total,  15149.6 libr,   2083.0 util,   6650.9 tamp/cache
MiB Éch:   2048.0 total,   2048.0 libr,      0.0 util.  21238.0 dispo Mem

    PID UTIL.      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TEMPS+ COM.
   1952 bilel      20   0 3429584 183432  98808 S  16.3   0.8   2:59.86 cinnamon
   1128 root       20   0  923600 309028 113816 S   5.0   1.3   8:10.10 Xorg
  13415 bilel      20   0  684824  46580  34444 S   2.3   0.2   0:00.98 gnome-screensho
   1672 bilel       9 -11 2304648  21032  16316 S   2.0   0.1  13:59.84 pulseaudio
   1143 root      -51   0       0      0      0 S   1.3   0.0   1:04.67 irq/142-nvidia
   1145 root       20   0       0      0      0 S   0.3   0.0   0:16.78 nv_queue
   5485 bilel      20   0 2816908 269688 143020 S   0.3   1.1   4:06.24 Web Content
   5903 bilel      20   0 2622880 184888 111076 S   0.3   0.8   5:49.55 Web Content
  13529 root       20   0       0      0      0 I   0.3   0.0   0:00.04 kworker/u16:3-events_power_efficient
  13650 bilel      20   0   14900   4304   3468 R   0.3   0.0   0:00.07 top
      1 root       20   0  167952  11928   8452 S   0.0   0.0   0:01.68 systemd
      2 root       20   0       0      0      0 S   0.0   0.0   0:00.01 kthreadd
      3 root        0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root        0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      6 root        0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
      7 root       20   0       0      0      0 I   0.0   0.0   0:02.71 kworker/0:1-events
      9 root        0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
```

Commande Linux : I-Gestion des processus sous Linux

top est une application simple. Après l'exécution de la commande, le terminal change d'affichage. La liste des processus est constamment mise à jour toutes les 5 secondes environ. Ce nouvel affichage est interactif grâce à l'utilisation du clavier. Quelques exemples:

- **H ou ?**: Afficher une fenêtre d' aide avec toutes les commandes et autres informations utiles.
- **Espace**: Appuyez dessus pour mettre à jour la liste des processus.
- **F** : Ajouter des champs ou supprime certains champs.
- **Q** :quitte l'application top ou une fenêtre rattachée à top
- **L** : Affiche les informations relatives à la disponibilité et l'utilisation moyenne.
- **M** : Permet d'afficher des informations sur la mémoire.
- **P** (Shift + p) : Trier les processus en fonction de l'utilisation du processeur.

Autres usages utiles de top :

Pour afficher les processus d'un utilisateur en particulier, vous pouvez utiliser: top -u utilisateur

Pour tuer/arrêter un processus en cours d'exécution, trouvez le PID du processus que vous souhaitez tuer et appuyez sur k (une autre commande du clavier). Vous serez invité à entrer ce même PID et à exécuter la commande.

Commande Linux : I-Gestion des processus sous Linux

Ps : Une autre commande très utile pour afficher les processus sous Linux. Voici quelques options fréquemment utilisées avec la commande ps :

-e : Affiche tous les processus.

-f : Listing complet.

-r : Affiche uniquement les processus en cours d'exécution.

-u : Possibilité d'utiliser un nom d'utilisateur (ou plusieurs) en particulier.

-pid : Option de filtrage par PID

-ppid :Option de filtrage par PPID

ps -ef – répertorie les processus en cours d'exécution. (Une autre commande similaire est ps aux)

ps -f -u user1,user2 – Affiche tous les processus basés sur un ou des UID en particulier (User ID ou nom d'utilisateur).

ps aux --sort=-pcpu,+pmem – Affiche les processus consommant la plus grande quantité de CPU.

ps -e -o pid,uname,pcpu,pmem,comm – Utilisé pour afficher certaines colonnes seulement.

ps -e -o pid,comm,etime – Affiche le temps depuis lequel le processus a démarré.

Nous vous recommandons de consulter la page aide « man ps » pour plus d'informations et l'utilisation de la commande ps .

Commande Linux : I-Gestion des processus sous Linux

2. Tuer et hiérarchiser les processus

Nous vous avons déjà montré comment tuer/arrêter un processus avec la commande top . Mais vous pouvez également le faire avec la commande **kill** . Par exemple:

kill pid : Ici, au lieu du PID, vous devez entrer l'ID du processus que vous voulez tuer. Si le processus ne veut pas s'arrêter, vous pouvez utiliser: **kill -9 pid** .

Une autre commande utile pour la gestion des processus est NICE . Il vous permet de hiérarchiser les processus au cas où vous en exécutez beaucoup sur votre système. De cette façon, votre système saura quels processus sont les plus importants et les exécutera en premier. Cette commande vous aide à prioriser les processus du plus important au moins important. Pour les processus qui ont une priorité inférieure, le système ne les exécutera que s'il le peut (si une le processeur a assez de ressources à y allouer). Cette commande peut recevoir une valeur entre -20 et 19. Plus la valeur est basse, plus la priorité sera élevée pour un processus. La priorité par défaut de TOUS les processus est 0. La syntaxe de base est la suivante:

nice -n 'Priorité' processus nom – Exemple: **nice -n 10 nom** . Cela démarrera un nouveau processus avec la priorité 10.

S'il existe déjà un processus s'exécutant sur le système avec le même nom et que vous voulez lui donner une priorité différente, vous pouvez utiliser:

renice 'Priorité' -p 'PID' Exemple: **renice '10' -p '54125'** .

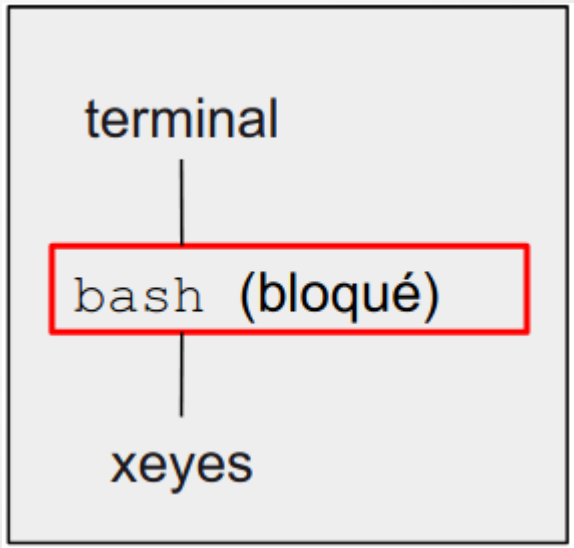
Commande Linux : I-Gestion des processus sous Linux

3. Contrôle des tâches

Dans un processus bash, il est possible de démarrer plusieurs processus appelés aussi jobs. Par défaut, un processus est démarré en avant-plan et il est le seul à recevoir les données de l'entrée standard (le clavier). Il faut utiliser Ctrl+Z pour le suspendre ou Ctrl+C pour l'arrêter. Pour démarrer un processus en arrière-plan, il faut utiliser le signe « **&** ».

Par défaut, une commande s'exécute en avant-plan (en anglais, foreground)

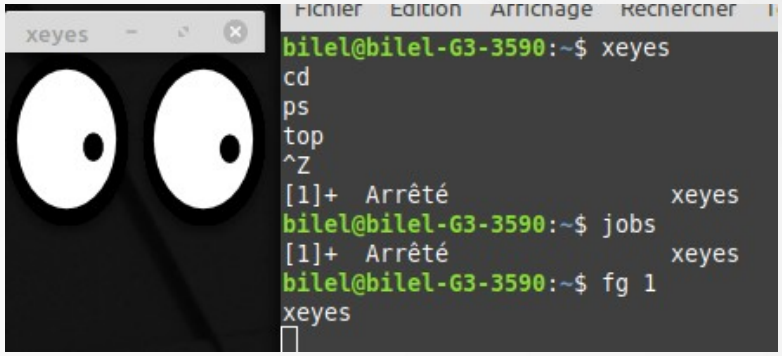
- bash crée un processus enfant et attend qu'il termine
- Le processus enfant exécute le programme
- bash est bloqué tant que le processus fils s'exécute



Commande Linux : I-Gestion des processus sous Linux

1-Suspendre un job, taper Ctrl+Z, le job passe dans un état arrêté
La commande jobs : lister tous les jobs

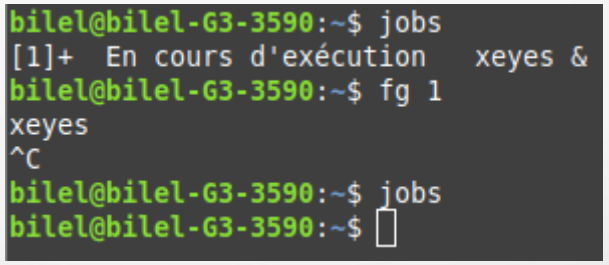
2- la commande fg reprendre la processus



3- j'ai fait Ctrl+Z : le processus est suspendu
bg : Continuer l'exécution d'un job tournant en arrière-plan

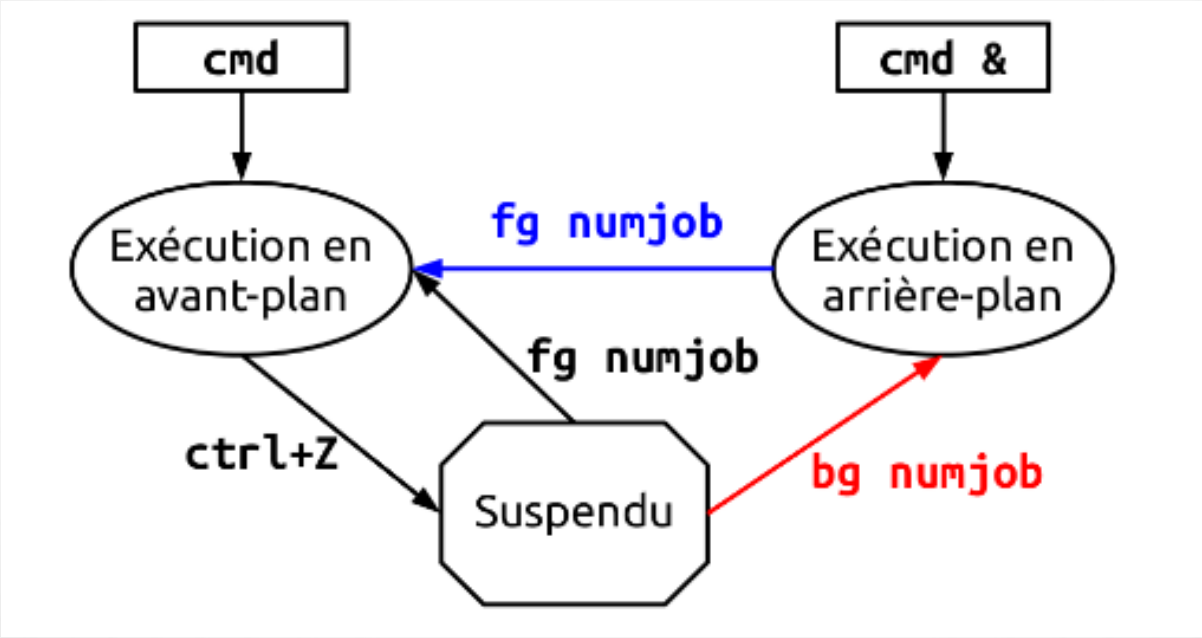
4- fg : Ramener un job en avant-plan

Ctrl+C : arrêter le processus



démarrer un processus en arrière-plan avec le signe &

Commande Linux : I-Gestion des processus sous Linux



Commande Linux : II-administration réseaux

1-Les fichiers de configuration

a-Le fichier /etc/hosts : Le fichier hosts donne un moyen d'assurer la résolution de noms, de donner un nom FQDN à un hôte

```
bilel@bilel-G3-3590:~$ cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       bilel-G3-3590
192.168.1.186   bilel.test myapp.test k8s
192.168.1.166 jenkins.test jenkins
```

b-Le fichier /etc/networks : Il permet d'affecter un nom logique à un réseau

```
bilel@bilel-G3-3590:~$ cat /etc/networks
# symbolic names for networks, see networks(5) for more information
link-local 192.168.1.0
```

Cette option permet par exemple d'adresser un réseau sur son nom, plutôt que sur son adresse.

route add dep-devops au lieu de route **add -net 192.168.0.0**.

```
bilel@bilel-G3-3590:~$ route
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    100    0        0 enp3s0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 enp3s0
172.17.0.0      0.0.0.0         255.255.0.0     U     0      0        0 docker0
link-local      0.0.0.0         255.255.255.0   U     100    0        0 enp3s0
```

Commande Linux : II-administration réseaux

c-Le fichier /etc/host.conf : Il donne l'ordre dans lequel le processus de résolution de noms est effectué.

```
bilel@bilel-G3-3590:~$ cat /etc/host.conf
# The "order" line is only used by old versions of the C library.
order hosts,bind
multi on
```

La résolution est effectuée d'abord avec le fichier hosts, en cas d'échec avec le DNS.

d-Le fichier /etc/resolv.conf : Il permet d'affecter les serveurs de noms.

```
Nameserver 192.168.1.1
Nameserver 192.168.1.2
Nameserver 192.168.1.3
```

Ici le fichier déclare le nom de domaine et les 3 machines chargées de la résolution de noms.

e-Le fichier /etc/network/interfaces : fichier de configuration des interfaces réseau

```
bilel@bilel-G3-3590:~$ cat /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo eth0
iface lo inet loopback
iface eth0 inet static
        address 192.168.1.2
        netmask 255.255.255.0
        network 192.168.1.0
        broadcast 192.168.1.255
        gateway 192.168.90.1
bilel@bilel-G3-3590:~$
```

Commande Linux : II-administration réseaux

2-Les outils de l'administration réseaux

a-La commande ifconfig :permet d'afficher les paramètres réseau des interfaces.

```
bilel@bilel-G3-3590:~$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:9f:9e:d9:83  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.16  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::ab4d:2111:555c:654d  prefixlen 64  scopeid 0x20<link>
        ether e4:54:e8:4b:ab:f4  txqueuelen 1000  (Ethernet)
        RX packets 288458  bytes 143992609 (143.9 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 78244  bytes 9857989 (9.8 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Boucle locale)
        RX packets 9538  bytes 848228 (848.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9538  bytes 848228 (848.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

bilel@bilel-G3-3590:~$
```

b- la commande ip : permettant d'afficher et manipuler le routage, les périphériques réseaux et les interfaces.

Ip a : Afficher des informations sur toutes les interfaces réseau

```
bilel@bilel-G3-3590:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether e4:54:e8:4b:ab:f4 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.16/24 brd 192.168.1.255 scope global dynamic noprefixroute enp3s0
       valid_lft 80609sec preferred_lft 80609sec
    inet6 fe80::ab4d:2111:555c:654d/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: wlo1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 08:71:90:12:26:68 brd ff:ff:ff:ff:ff:ff
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:9f:9e:d9:83 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
bilel@bilel-G3-3590:~$
```

Commande Linux : II-administration réseaux

Affichage des information avec la commande ip

```
### Afficher des informations sur toutes les interfaces réseau
ip a

### N'afficher que de l'IPv4
ip  -4 a
### N'afficher que de l'IPv6
ip  -6 a

### Ne montre que l'interface ens3
ip a show ens3

## Affiche uniquement les interfaces à l'état UP
ip link ls up
```

Assigner/Supprimer une adresse IP à une interface réseau

```
### Assigner l'adresse 192.168.1.3 avec le masque sous réseau 255.255.255.0 à l'interface ens3
ip a add 192.168.1.3/255.255.255.0 dev ens3
## ou
ip a add 192.168.1.3/24 dev ens3

### Supprimer l'adresse IP 192.168.1. de l'interface ens3
ip a del 192.168.1.3/24 dev ens3
```

Changer l'état d'une interface en UP ou DOWN

```
### Désactiver l’état du périphérique ens3
ip link set dev ens3 down

### Rétablir l’état du périphérique ens3
ip link set dev ens3 up
```

Commande Linux : II-administration réseaux

c-La commande ping : permet de tester la connectivité entre deux systèmes sur un réseau local (LAN) ou un réseau étendu (WAN).

Pour information cette commande utilise le protocole ICMP (Internet Control Message Protocol) pour communiquer avec les nœuds d'un réseau.

Vous indiquerez dans la commande simplement une adresse IP ou un nom d'hôte :

```
bilel@bilel-G3-3590:~$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=30 time=0.509 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=30 time=0.598 ms
^C
--- 192.168.1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.509/0.553/0.598/0.044 ms
bilel@bilel-G3-3590:~$ ping google.com
PING google.com (216.58.205.206) 56(84) bytes of data.
64 bytes from mil04s29-in-f206.1e100.net (216.58.205.206): icmp_seq=1 ttl=118 time=39.0 ms
64 bytes from mil04s29-in-f206.1e100.net (216.58.205.206): icmp_seq=2 ttl=118 time=31.4 ms
64 bytes from mil04s29-in-f206.1e100.net (216.58.205.206): icmp_seq=3 ttl=118 time=31.4 ms
^C
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 31.370/33.912/38.954/3.565 ms
bilel@bilel-G3-3590:~$
```

Commande Linux : II-administration réseaux

d-La commande ARP : permet de traduire les adresses IP en adresses physique.

```
bilel@bilel-G3-3590:~$ arp 192.168.1.1
Adresse                  TypeMap AdresseMat          Indicateurs          Iface
_gateway                 ether   34:e8:94:ed:0e:38   C                    enp3s0
bilel@bilel-G3-3590:~$
```

Afficher la table ARP afin de connaître les adresses MAC des machines de votre réseau :

```
bilel@bilel-G3-3590:~$ arp
Adresse                  TypeMap AdresseMat          Indicateurs          Iface
_gateway                 ether   34:e8:94:ed:0e:38   C                    enp3s0
192.168.1.12             ether   94:a1:a2:3e:02:40   C                    enp3s0
bilel@bilel-G3-3590:~$
```

e-La commande traceroute : permet de découvrir la source de blocage d'un paquet, puisqu'elle permet de suivre le chemin complet de votre système local à un autre système réseau. Elle affiche le nombre de sauts (adresses IP du routeur) dans le chemin emprunté pour atteindre le serveur final.

```
bilel@bilel-G3-3590:~$ traceroute google.com
traceroute to google.com (172.217.18.238), 64 hops max
  1   192.168.1.1  0.583ms  0.553ms  0.538ms
  2   41.226.21.227  16.652ms  172.26.0.2  16.137ms  15.971ms
  3   172.26.0.2  16.851ms  193.95.96.98  16.201ms  15.907ms
  4   193.95.96.98  19.715ms  16.249ms  15.877ms
  5   193.95.96.98  19.785ms  193.95.0.150  16.471ms  16.978ms
  6   193.95.0.150  16.575ms  193.95.1.221  22.915ms  23.081ms
  7   193.95.1.221  16.236ms  72.14.194.136  31.302ms  31.267ms
  8   72.14.194.136  31.453ms  108.170.252.225  33.095ms  31.963ms
  9   108.170.252.225  32.519ms  72.14.232.43  32.110ms  32.112ms
 10   72.14.232.49  32.811ms  172.217.18.238  30.865ms  31.762ms
bilel@bilel-G3-3590:~$
```

Commande Linux : II-administration réseaux

f-La commande route : permet d'afficher ou de manipuler la table de routage IP d'un système Linux. Elle est principalement utilisée pour définir un chemin de route statique dans les tables de route. Voici mes différents cas d'utilisation.

```
bilel@bilel-G3-3590:~$ route
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    100    0        0 enp3s0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 enp3s0
172.17.0.0      0.0.0.0         255.255.0.0     U     0      0        0 docker0
link-local      0.0.0.0         255.255.255.0   U     100    0        0 enp3s0
```

Ajouter un itinéraire réseau à la table de routage

```
$ sudo route add -net 192.168.1.3 netmask 255.255.255.0 gw 192.168.1.1 dev eth0
## ou
sudo route add -net 192.168.1.3/24 gw 192.168.1.1 dev eth0
```

Supprimer une entrée de route spécifique de la table de routage

```
$ sudo route del -net 192.168.1.3/24 gw 192.168.1.1 dev eth0
```

Commande Linux : II-administration réseaux

g-La commande nslookup : permet d'interroger le serveur DNS dans le but de traduire une adresse IP en un nom de domaine, ou inversement.

```
bilel@bilel-G3-3590:~$ nslookup google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 172.217.18.46
Name:   google.com
Address: 2a00:1450:4006:804::200e
```

h-La commande dig : permet de interroger des informations liées au DNS telles que l'enregistrement A, CNAME, l'enregistrement MX, etc.

```
bilel@bilel-G3-3590:~$ dig ghazelatc.com

; <<>> DiG 9.16.1-Ubuntu <<>> ghazelatc.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54200
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;ghazelatc.com.                 IN      A

;; ANSWER SECTION:
ghazelatc.com.          3582    IN      A       51.255.194.100

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: sam. oct. 17 11:43:34 CET 2020
;; MSG SIZE  rcvd: 58
```

Commande Linux : II-administration réseaux

i-La commande netstat : permet d'examiner chacune de mes connexions réseau et de mes sockets ouverts.

```
bilel@bilel-G3-3590:~$ netstat -lntp
(Tous les processus ne peuvent être identifiés, les infos sur les processus
non possédés ne seront pas affichées, vous devez être root pour les voir toutes.)
Connexions Internet actives (seulement serveurs)
Proto Recv-Q Send-Q Adresse locale          Adresse distante        Etat       PID/Program name
tcp        0      0 127.0.0.1:5939          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:2049            0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:41989           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:46093           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:43535           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:43599           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      -
```

Par exemple, la commande suivante affiche tous les ports TCP en mode d'écoute et les programmes en cours d'écoute.

L'état LISTEN signifie que le programme écoute et attend une connexion, mais vous pourriez aussi avoir l'état ESTABLISHED lorsqu'une connexion est déjà établie.

Remplacez l'option -t par un -u pour examiner les ports UDP. **Netstat -lnup**

Par défaut, les statistiques sont affichées pour les protocoles TCP, UDP, ICMP et IP. Le paramètre -s peut être utilisé pour spécifier cet ensemble de protocoles.

```
### Statistique de tous les protocoles
netstat -s
### Statistique que pour le protocole TCP
netstat -st
### Statistique que pour le protocole UDP
netstat -su
```

Commande Linux : II-administration réseaux

j-La commande nmap : permet de vérifier les ports ouvert sur un serveur.

```
bilel@bilel-G3-3590:~$ nmap google.com
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-17 12:16 CET
Nmap scan report for google.com (172.217.19.142)
Host is up (0.033s latency).
Other addresses for google.com (not scanned): 2a00:1450:4006:801::200e
rDNS record for 172.217.19.142: par03s12-in-f142.1e100.net
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 6.64 seconds
```

```
bilel@bilel-G3-3590:~$ nmap ghazelatc.com
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-17 12:16 CET
Nmap scan report for ghazelatc.com (51.255.194.100)
Host is up (0.74s latency).
rDNS record for 51.255.194.100: 100.ip-51-255-194.eu
Not shown: 997 filtered ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 71.56 seconds
```

On peut aussi scanner une plage d'IP :

```
### Scanner par masque sous réseau (va scanner de l'ip 192.168.1.0 à 192.168.1.255)
nmap 192.168.1.0/24

### Scanner la plage IP de 192.168.1.1 à 192.168.1.200
nmap 192.168.1.1-200
```

Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000.

Commande Linux : II-administration réseaux

k-La commande tcpdump : utilisé pour capturer et analyser les paquets TCP/IP transmis ou reçus sur un réseau ou sur une interface spécifique.

Voici quelques options a utiliser :

-w : capturer les paquets dans un fichier qui pourra être analysé plus tard

-i : utiliser une interface réseau

-A : Voir le contenu d'un paquet IP

port : Filtrage par port

host : adresse de destination et/ou source

dst : adresse de destination

src : adresse source

-r: lire le paquet capturer

Ci-dessous quelques exemples d'utilisation :

```
### écouter le port http (80) sur l'interface ens3 et voir le contenu du
tcpdump -A -i ens3 port http

### Affiche tous les paquets en provenance de 192.168.1.2 vers 192.168.1.3 sur le port 22 en tcp.
tcpdump src host 192.168.1.2 and dst host 192.168.1.3 and port 22 and tcp

### stocker la capture dans le fichier capture.tdp
tcpdump -w capture.tdp

### lire la capture dans le fichier capture.tdp
tcpdump -v -r capture.tdp
```

Commande Linux : II-administration réseaux

L- UFW : est un nouvel outil de configuration simplifié en ligne de commande qui est une alternative à l'outil iptables. Il est par défaut sur les distributions Debian et Ubuntu Linux et est utilisé pour ajouter/supprimer/modifier/réinitialiser les règles de filtrage de paquets du pare-feu de votre système.

Avant de rajouter des règles, il faut d'abord vérifier le statut de l'outil UFW à l'aide de la commande suivante:

```
$ sudo ufw status

Status: active
```

S'il n'est pas activé alors lancez la commande suivante :

```
$ sudo ufw enable
```

Par défaut, UFW bloquera toutes les connexions entrantes et autorisera toutes les connexions sortantes. Cela signifie que toute personne essayant d'accéder à votre serveur ne pourra pas se connecter à moins que vous n'ouvriez spécifiquement un port, tandis que toutes les applications et tous les services exécutés sur votre serveur pourront accéder au monde extérieur.

Autoriser et refuser les connexions :

```
### Autoriser un protocole ou une ip
sudo ufw allow ssh
sudo ufw allow 80/tcp

### Autoriser l'accès en sortie à un serveur ssh
ufw allow out 22/tcp


### Autoriser l'accès en entré (de l'extérieur) à notre serveur en ssh
ufw allow in 22/tcp

## Seul l'ip 192.168.1.3 est autorisée à accéder à notre serveur sur le port 5876
ufw allow from 192.168.1.3 to any port 5876

### Autoriser une plage de port
sudo ufw allow 1000:2000/tcp
```

Commande Linux : II-administration réseaux

N'oubliez pas de charger vos nouvelles règles avec la commande ci-dessous : `$ ufw reload`

Voici la commande pour vérifier l'état actuel de votre firewall :

```
$ ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), deny (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
22                         ALLOW IN    Anywhere
22 (v6)                    ALLOW IN    Anywhere (v6)
```

Chaque règle possède un numéro que vous pouvez lister avec la commande suivante :

```
$ ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22                         ALLOW IN    Anywhere
[ 2] 22 (v6)                    ALLOW IN    Anywhere (v6)
```

Supprimer simplement une règle d'après son numéro : `$ sudo ufw delete [numéro]`

Commande Linux : III- gestion des services

Les services permettent de démarrer automatiquement des programmes lors du démarrage du système d'exploitation comme un serveur de base de donner Mysql-Server ou serveur web apache2.

On va explique comment créer un service personnalisé s'exécutant sous Systemd sous Linux. De cette façon, vous pourrez utiliser la même commande que vous utilisez pour gérer un service Apache ou Nginx par exemple, mais cette fois-ci pour gérer vos propres services.

Fichiers de configuration

La configuration des services se trouve par défaut dans le répertoire /lib/systemd/system (Ubuntu, Linuxmint) ou /usr/lib/systemd/system (centos).

On utilisera le répertoire /etc/systemd/system pour stocker ses modifications et configurations personnelles, ce qui a le gros avantage que ces fichiers ne seront pas effacés en cas de mise à jour du système.

Les unités

La configuration de Systemd se base sur des unités (units) qui ont un nom et un type. Ainsi, le fichier NetworkManager.service définira l'unité de type service qui s'occupe de la gestion réseau.

La commande suivante vous listera les unités disponibles sur votre système : **systemctl -t help**

```
bilel@bilel-G3-3590:/lib/systemd/system$ systemctl -t help
Available unit types:
service
mount
swap
socket
target
device
automount
timer
path
slice
scope
bilel@bilel-G3-3590:/lib/systemd/system$
```

Commande Linux : III- gestion des services

Les principaux différents types sont :

- **service :** pour un service système
- **mount :** pour un système de fichiers (exemple : home.mount), tout en utilisant /etc/fstab
- **swap :** pour les partitions de swap
- **socket :** pour une socket de communication entre processus (de tous types : UNIX, Internet, fichier etc.)
- **target :** macro-unité qui permet de grouper plusieurs unités (exemple : multi-user.target pour définir une cible)
- **device :** pour un périphérique
- **automount :** pour un système de fichiers monté à la demande
- **timer :** pour l'activation basée sur une date
- **path :** pour l'activation d'un service basée sur la modification de fichiers ou de répertoires
- **slice :** sert pour la gestion des cgroups, une fonctionnalité du noyau Linux pour limiter, compter et isoler l'utilisation des ressources
- **scope :** utilisé par systemd lui-même pour gérer des groupes de processus, typiquement, par session utilisateurs ;

Pour lister toutes les unités présentes sur le système, on fera : **systemctl list-units**

Commande Linux : III- gestion des services

```
bilel@bilel-G3-3590:/lib/systemd/system$ systemctl list-units
  UNIT                                                                      >
  proc-sys-fs-binfmt_misc.automount                                         >
  sys-bus-pci-drivers-nvidia.device                                         >
  sys-devices-pci0000:00-0000:00:01.0-0000:01:00.1-sound-card0.device       >
  sys-devices-pci0000:00-0000:00:02.0-drm-card0-card0\x2deDP\x2d1-intel_backlig>
  sys-devices-pci0000:00-0000:00:14.0-usb1-1\x2d14-1\x2d14:1.0-bluetooth-hci0.d>
  sys-devices-pci0000:00-0000:00:14.3-net-wlo1.device                       >
  sys-devices-pci0000:00-0000:00:17.0-ata1-host0-target0:0:0-0:0:0:0-block-sda->
  sys-devices-pci0000:00-0000:00:17.0-ata1-host0-target0:0:0-0:0:0:0-block-sda->
  sys-devices-pci0000:00-0000:00:17.0-ata1-host0-target0:0:0-0:0:0:0-block-sda.>
  sys-devices-pci0000:00-0000:00:1d.0-0000:02:00.0-ata2-host1-target1:0:0-1:0:0>
  sys-devices-pci0000:00-0000:00:1d.0-0000:02:00.0-ata2-host1-target1:0:0-1:0:0>
  sys-devices-pci0000:00-0000:00:1d.0-0000:02:00.0-ata2-host1-target1:0:0-1:0:0>
  sys-devices-pci0000:00-0000:00:1d.0-0000:02:00.0-ata2-host1-target1:0:0-1:0:0>
  sys-devices-pci0000:00-0000:00:1d.0-0000:02:00.0-ata2-host1-target1:0:0-1:0:0>
  sys-devices-pci0000:00-0000:00:1d.0-0000:02:00.0-ata2-host1-target1:0:0-1:0:0>
  sys-devices-pci0000:00-0000:00:1d.0-0000:02:00.0-ata2-host1-target1:0:0-1:0:0>
  sys-devices-pci0000:00-0000:00:1d.0-0000:02:00.0-ata2-host1-target1:0:0-1:0:0>
  sys-devices-pci0000:00-0000:00:1d.5-0000:03:00.0-net-enp3s0.device        >
  sys-devices-pci0000:00-0000:00:1f.3-skl_hda_dsp_generic-sound-card1.device >
  sys-devices-platform-dell\x2dlaptop-leds-dell::kbd_backlight.device       >
```

Commande Linux : III- gestion des services

Gestion des services

Pour faire une action sur un service, c'est facile, on fera : systemctl <action> <nom_du_service>.service.

Par exemple pour le service NetworkManager, on fera des commandes du type : **systemctl start NetworkManager.service**

Démarrer, arrêter un service

Dans les exemples ci-dessous, remplacez application par le nom de votre service : par exemple, mariadb, httpd, firewalld, nfsd etc.

-Pour démarrer le service :
systemctl start application.service

-Pour arrêter le service :
systemctl stop application.service

-Pour redémarrer un service qui est lancé, faites :
sudo systemctl restart application.service

-Pour recharger les fichiers de configuration d'un service dans le redémarrer (typiquement, le serveur web Apache) , faites :
sudo systemctl reload application.service

-Pour que le service soit lancé au démarrage du système :
systemctl enable application.service

-Pour que le service ne soit pas lancé au démarrage du système :
systemctl disable application.service

-Pour empêcher l'activation d'un service (par exemple on masquera httpd car on veut utiliser nginx)
systemctl mask application.service

-Pour envoyer un signal d'arrêt (SIGTERM) à tous les processus du service (plus élégant qu'un killall qui tue en fonction d'une chaine de caractère):
systemctl kill application.service

Commande Linux : III- gestion des services

Information sur un service

-Pour voir tous les services disponibles et leur statut, y compris les services de statut inactif :
systemctl list-unit-files --type=service --all

-Pour vérifier si le service est démarré, arrêté, afficher des informations pour le niveau d'exécution en cours :
systemctl status application.service

-Pour voir si le service est actuellement démarré :
systemctl is-active application.service

-Pour vérifier si le service sera démarré au démarrage du système :
systemctl is-enabled application.service

-Pou vérifier qu'il y a eu un problème lors du démarrage d'un service :
systemctl is-failed application.service

-Pour voir tous les services qui ont un problème :
systemctl --failed --type=service

-Évidemment les commandes Unix classiques vous permettent aussi de comprendre ce qui se passe, vous pouvez par exemple lister les processus avec :
ps aux

Commande Linux : III- gestion des services

Créer son propre service : exemple je vais crée un service backup.

1- crée scripte **backup.sh**

2- changez ses droits d'exécution

3- crée un service **backup.service** de type simple

```
bilel@bilel-G3-3590:~$ nano backup.sh
bilel@bilel-G3-3590:~$ cat backup.sh
#!/bin/bash
echo "bonjour backup scripte"
touch /mnt/hello_from_scripte_systemd
bilel@bilel-G3-3590:~$ chmod a+x backup.sh
bilel@bilel-G3-3590:~$ cd /etc/systemd/system
bilel@bilel-G3-3590:/etc/systemd/system$ nano backup.service
bilel@bilel-G3-3590:/etc/systemd/system$ sudo nano backup.service
[sudo] Mot de passe de bilel :
bilel@bilel-G3-3590:/etc/systemd/system$ cat backup.service
[Unit]
Description=Service de backup
[Service]
Type=simple
ExecStart=/home/bilel/backup.sh
[Install]
WantedBy=multi-user.target
bilel@bilel-G3-3590:/etc/systemd/system$
```

Commande Linux : III- gestion des services

4- daemon-reload Recharger la configuration du gestionnaire systemd
5- Démarrer le service
6- Activer le service au démarrage
7- vérifier le status du service

```
bilel@bilel-G3-3590:/etc/systemd/system$ sudo systemctl daemon-reload
bilel@bilel-G3-3590:/etc/systemd/system$ sudo systemctl start backup.service
bilel@bilel-G3-3590:/etc/systemd/system$ sudo systemctl enable backup.service
Removed /etc/systemd/system/multi-user.target.wants/backup.service.
Created symlink /etc/systemd/system/multi-user.target.wants/backup.service → /etc/systemd/system/backup.service.
bilel@bilel-G3-3590:/etc/systemd/system$ sudo systemctl status backup.service
● backup.service - Service de backup
     Loaded: loaded (/etc/systemd/system/backup.service; enabled; vendor preset: enabled)
     Active: inactive (dead) since Sat 2020-10-17 22:52:20 CET; 20s ago
   Main PID: 45885 (code=exited, status=0/SUCCESS)

oct. 17 22:52:20 bilel-G3-3590 systemd[1]: Started Service de backup.
oct. 17 22:52:20 bilel-G3-3590 backup.sh[45885]: bonjour backup scripte
oct. 17 22:52:20 bilel-G3-3590 systemd[1]: backup.service: Succeeded.
bilel@bilel-G3-3590:/etc/systemd/system$
```