Leçon 5 : Configuration réseau et démarrage / arrêt du système

5.1 Modifier la configuration réseau

5.1.1 Les interfaces réseaux

Une *interface réseau* dans les systèmes Linux est une *carte réseau* physique ou virtuelle, qui est présente sur le système. Les interfaces réseaux sont nommées de manière unique afin de pouvoir configurer l'environnement réseau complètement.

Ainsi, sur certains systèmes Linux, les interfaces sont nommées `eth0`, `eth1`, … (pour ethernet) ou encore `em0`, `em1`, …. Le nom des interfaces évolue beaucoup suivant la version du noyau Linux utilisé (afin de permettre une identification claire et non-ambigüe). Sur les systèmes CentOS 7, le nom des interfaces ethernet commencent par `eno` (`en` pour ethernet, `o` pour périphérique *on-board* et ensuite, un numéro identifiant provenant du matériel). Ainsi, l'interface réseau installée sur la machine se nomme `eno16777736`.

Les outils pour gérer les interfaces réseaux ont également beaucoup évolués. Sur tous les systèmes, on trouve les commandes classiques `ifconfig`, `netstat` ou `route`. Sur les systèmes plus récents, en plus de continuer à supporter les commandes *classiques*, la nouvelle commande `ip` est présente. Dans la suite du cours, les deux versions (anciennes et nouvelles) seront présentées.

Il est possible de visualiser les interfaces réseaux actives en utilisant les commandes `ip` suivantes :

```
[root@localhost Desktop]# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    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: eno16777736: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP qlen 1000
    link/ether 00:0c:29:56:25:af brd ff:ff:ff:ff:ff:ff
    inet 192.168.190.101/24 brd 192.168.190.255 scope global dynamic eno16777736
       valid_lft 6555sec preferred_lft 6555sec
    inet6 fe80::20c:29ff:fe56:25af/64 scope link
       valid_lft forever preferred_lft forever
```

Cette commande affiche les interfaces (`lo` et `eno16777736` dans notre exemple), les informations sur celles-ci et les configurations réseaux associées.

Pour visualiser uniquement les informations au niveau de la couche liaison de données :

```
[root@localhost Desktop]# ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eno16777736: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP mode DEFAULT qlen 1000
    link/ether 00:0c:29:56:25:af brd ff:ff:ff:ff:ff:ff
```

Avec la commande `ifconfig`, il est possible de visualiser ces mêmes informations :

```
[root@localhost Desktop]# ifconfig

eno16777736: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.190.101  netmask 255.255.255.0  broadcast 192.168.190.255
        inet6 fe80::20c:29ff:fe56:25af  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:56:25:af  txqueuelen 1000  (Ethernet)
        RX packets 125  bytes 14556 (14.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 83  bytes 10328 (10.0 KiB)
        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 0  (Local Loopback)
        RX packets 727  bytes 98958 (96.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 727  bytes 98958 (96.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Cette commande affiche la configuration des différentes interfaces réseaux actives. Avec l'option `-a`, les interfaces inactives ou non-configurées apparaissent également.

Grâce à ces commandes, il est également **possible de modifier la configuration courante d'une interface** (cela signifie qu'au prochain redémarrage, la configuration d'origine, sauvegardée, sera à nouveau installée.

Ainsi, avec la commande `ip`, il faut supprimer l'adresse précédente et en ajouter une nouvelle :

```
$ ip addr del 192.168.190.101/24 dev eno16777736
$ ip addr add 10.0.1.2/24 dev eno16777736
```

La commande `ifconfig` suivante modifie également la configuration courante de l'interface :

```
$ ifconfig eno16777736 10.0.1.2 netmask 255.255.255.0
```

Cette commande affecte l'adresse 10.0.1.2 à l'interface eno16777736 avec un masque /24.

Pour afficher **la table de routage** de la machine, nous avons les deux commandes suivantes :

```
[root@localhost Desktop]# ip route show
default via 192.168.190.2 dev eno16777736  proto static  metric 100
192.168.190.0/24 dev eno16777736  proto kernel  scope link  src
192.168.190.101  metric 100
```

```
[root@localhost Desktop]# netstat -r -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         192.168.190.2   0.0.0.0         UG        0 0          0 eno16777736
192.168.190.0   0.0.0.0         255.255.255.0   U         0 0          0 eno16777736
```

Dans les 2 cas, nous avons **la route par défaut** (appelée *default* ou *0.0.0.0*) qui utilise le relai `192.168.190.2` (adresse IP de pfSense) pour atteindre internet. Ce relai est connecté sur l'interface `eno16777736`. La seconde ligne mentionne que la machine **est directement connectée** (sans relai donc) au réseau `192.168.190.0/24` (aussi associé à l'interface `eno16777736`).

Modifier les paramètres réseaux configurés

Pour modifier définitivement la configuration réseau, nous pouvons utiliser différents moyens : *Webmin* peut être utilisé, tout comme un programme particulier, *NetworkManager Client* ou encore la modification manuelle des fichiers de configuration. Comme ces 3 méthodes permettent d'atteindre cet objectif, peu importe celle que vous décidez d'appliquer. Nous allons expliquer certaines d'entre-elles :

1. Via **Webmin**, il faut aller dans **Networking > Network Configuration**. Il faut ensuite cliquer sur **Network Interfaces** puis cliquer sur le nom de l'interface réseau à configurer (par exemple `eno16777736`). Il est alors possible d'entrer la configuration réseau souhaitée. Une fois les modifications terminées, il faut cliquer sur *Save and Apply* pour voir cette configuration devenir active.

   Attention ! Il faudra également configurer d'autres éléments réseaux importants comme la route par défaut (**Networking > Network Configuration** puis **Routing and Gateways**) et les serveurs DNS à interroger (**Networking > Network Configuration** puis **Hostname and DNS Client**).

2. En utilisant l'outil **Network Manager**. Cet outil permet de créer *des profils* contenant toutes la configuration réseau souhaitée. Ainsi, suivant l'emplacement, il est possible d'activer un ou l'autre profil. Il est clair que sur une configuration serveur, la plupart du temps, un seul profil sera utilisé (ce qui n'est pas le cas si nous avions installé CentOS sur un portable par exemple, ou nous pourrions avoir des profils différents pour le domicile ou le bureau). A nouveau, 2 outils sont présents : soit une version en ligne de commande, soit une version interactive :
   a. La version **en ligne de commande** est assez simple à utiliser. Nous pouvons commencer par lister les profils présents :

      ```
      [root@localhost Desktop]# nmcli device status
      DEVICE        TYPE      STATE      CONNECTION
      eno16777736   ethernet  connected  eno16777736
      lo            loopback  unmanaged  --
      ```

      Nous observons ici que l'interface *eno16777736* est associée à une connexion (= un profil) du même nom. Il est tout à fait possible de lire, adapter ou supprimer ce profil.

      Pour visualiser ce profil :

      ```
      [root@localhost Desktop]# nmcli -p connection show eno16777736
      ```

      Pour utiliser une adresse IP statique à la place du mode DHCP, nous utiliserons la commande suivante :

      ```
      [root@localhost Desktop]# nmcli connection modify eno16777736
      ipv4.method manual ipv4.address 10.0.1.2/24 ipv4.gateway
      10.0.1.1 ipv4.dns 8.8.8.8 ipv4.dns-search localdomain
      ```

      Cette commande modifie le profil eno16777736 pour utiliser l'adresse IP statique 10.0.1.2 avec le masque /24. La passerelle par défaut est également configurée à 10.0.1.1 et le serveur DNS à interroger est 8.8.8.8.

Pour activer les modifications, il est possible de redémarrer la machine ou simplement lancer les commandes :

```
$ nmcli device disconnect eno16777736
$ nmcli device connect eno16777736
```

Pour revenir à une configuration utilisant DHCP, il faut entrer :

```
[root@localhost Desktop]# nmcli connection modify eno16777736 ipv4.method auto
```

b. La version **interactive** peut être lancée par la commande :

```
$ nmtui
```

Cette commande lance un outil de configuration interactif qui permet de modifier une connexion.

Ainsi, l'option **Edit a connection** permet de modifier les profils présents (dans notre cas, eno16777736). Il est ensuite possible de modifier la configuration IPv4 (en basculant de *automatic* vers *manual*) et mentionner les paramètres réseaux à configurer.

3. En modifiant **les fichiers de configuration à la main** directement. Il faut commencer par le fichier `/etc/sysconfig/network-scripts/ifcfg-eno16777736` (pour une interface portant ce nom).

Ce fichier contient quelques lignes (les lignes en gras sont ajoutées ou modifiées, les lignes barrées sont supprimées) :

```
TYPE="Ethernet"
BOOTPROTO=none
NM_CONTROLLED="no"
DEFROUTE="yes"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_FAILURE_FATAL="no"
NAME="eno16777736"
UUID="0b673394-de1e-492d-bf8e-5d65da08581c"
DEVICE="eno16777736"
ONBOOT="yes"
~~PEERDNS=yes~~
~~PEERROUTES=yes~~
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPADDR=10.0.1.2
PREFIX=24
GATEWAY=10.0.1.1
DNS1=8.8.8.8
DOMAIN=localdomain
```

Une fois la modification terminée, il faut activer celle-ci en utilisant la commande :

```
$ systemctl restart network
```
Cette commande redémarre toute la configuration réseau.

Ajouter une 2ème adresse IP à une interface

Il est possible d'ajouter plusieurs adresses IP à la même interface. Ainsi les commandes suivantes (nouvelle et ancienne version) permettent de réaliser cet ajout de manière temporaire jusqu'au prochain redémarrage de la machine :

```
$ ip addr add 10.0.1.3/24 dev eno16777736
$ ifconfig eno16777736:1 10.0.1.3 netmask 255.255.255.0
```

Ainsi, la machine peut maintenant recevoir des informations provenant de toutes les adresses IP configurée. Dans le 1ier cas, nous ajoutons une nouvelle adresse à l'interface `eno16777736`. Dans le second cas, nous ajoutons une nouvelle interface, *virtuelle* et *liée à la première*, qui répond sur cette adresse IP.

La suppression de cette adresse se fait comme suit (suivant la méthode utilisée pour l'ajout) :

```
$ ip addr dele 10.0.1.3/24 dev eno16777736
$ ifconfig eno16777736:1 down
```

Pour **rendre définitif**[15] une seconde adresse IP, il est possible de passer par *Webmin, NetworkManager* ou le *fichier de configuration* :

1. Via **Webmin**, il faut aller dans **Networking > Network Configuration**. Il faut ensuite cliquer sur **Network Interfaces** puis cliquer sur le nom de l'interface réseau à configurer (par exemple `eno16777736`).

   Tout en dessous, nous avons *Virtual Interfaces*, il est alors possible de cliquer sur **Add virtual interface.** Dans l'écran suivant, il est possible de mentionner les paramètres pour cette interface virtuelle.
2. En utilisant **Network Manager,** nous pouvons réaliser l'opération :
   a. En ligne de commande :
      `$ nmcli connection modify eno16777736 +ipv4.address`[16]
      `10.0.1.3/24`

   b. Via l'outil interactif :
      `$ nmtui`
      Edit a connection puis **choisir le profil concerné** (par exemple eno16777736), et enfin, via l'option *Add* sous l'adresse IP déjà configurée, il est possible d'en ajouter une seconde.

3. En **modifiant les fichiers de configuration directement** : `/etc/sysconfig/network-scripts/ifcfg-eno16777736` (pour une interface portant ce nom). Pour ce faire, il faut **ajouter** les lignes suivantes :
   `IPADDR1=10.0.1.3`
   `PREFIX1=24`

[15] Il est nécessaire que l'adresse IP principale soit configurée statiquement (pas de DHCP)
[16] Pour supprimer cette configuration, il suffit de remplacer `+ipv4.address` par `-ipv4.address`

5.1.2 Désactiver / Redémarrer la configuration réseau

Nous avons, dans les exemples précédents, utilisés quelques commandes pour désactiver ou redémarrer la configuration réseau. Nous allons maintenant faire le point sur les différentes commandes permettant d'atteindre cet objectif.

En utilisant les outils *Network Manager* (pour autant que celui-ci soit utilisé), on peut désactiver le profil utilisé et le réactiver :

```
$ nmcli device disconnect eno16777736
$ nmcli device connect eno16777736
```

En utilisant les outils `ip` ou `ifconfig`, on peut désactiver l'interface et la réactiver :

```
$ ip link set eno16777736 down
$ ip link set eno16777736 up
$ ifconfig eno16777736 down
$ ifconfig eno16777736 up
```

Il est également possible de redémarrer la configuration complète du système linux comme suit :

```
$ systemctl restart network
```

5.2 Vérifier la configuration réseau

Comme nous venons de le remarquer, modifier la configuration est assez simple. Par contre, identifier le problème lorsqu'on est face à une configuration inadaptée n'est pas toujours aisé.

Comme nous pouvons le voir sur la figure 5.1, vérifier la configuration réseau peut se faire en suivant les étapes énoncées :

1. *Ping vers la passerelle par défaut.* L'objectif est de voir *si une autre machine directement connectée* à celle-ci répond aux requêtes (! Vérifiez bien qu'aucun firewall n'est actif !)
   Si la machine ne répond pas, il faut commencer par vérifier si la configuration réseau de base (adresse IP, masque, interface réseau, …) est correcte.
2. *Ping vers 8.8.8.8.* L'objectif est de tester s'il est possible de sortir du réseau. Si aucune réponse n'est obtenue, il faut vérifier si la connexion internet est bien disponible, si la passerelle utilisée est la bonne.
3. *Ping vers www.google.com ou www.yahoo.fr*. L'objectif est de voir si un nom DNS peut être résolu. Si aucune réponse n'est renvoyée, il faut contrôler le serveur DNS utilisé. Est-il fonctionnel ?

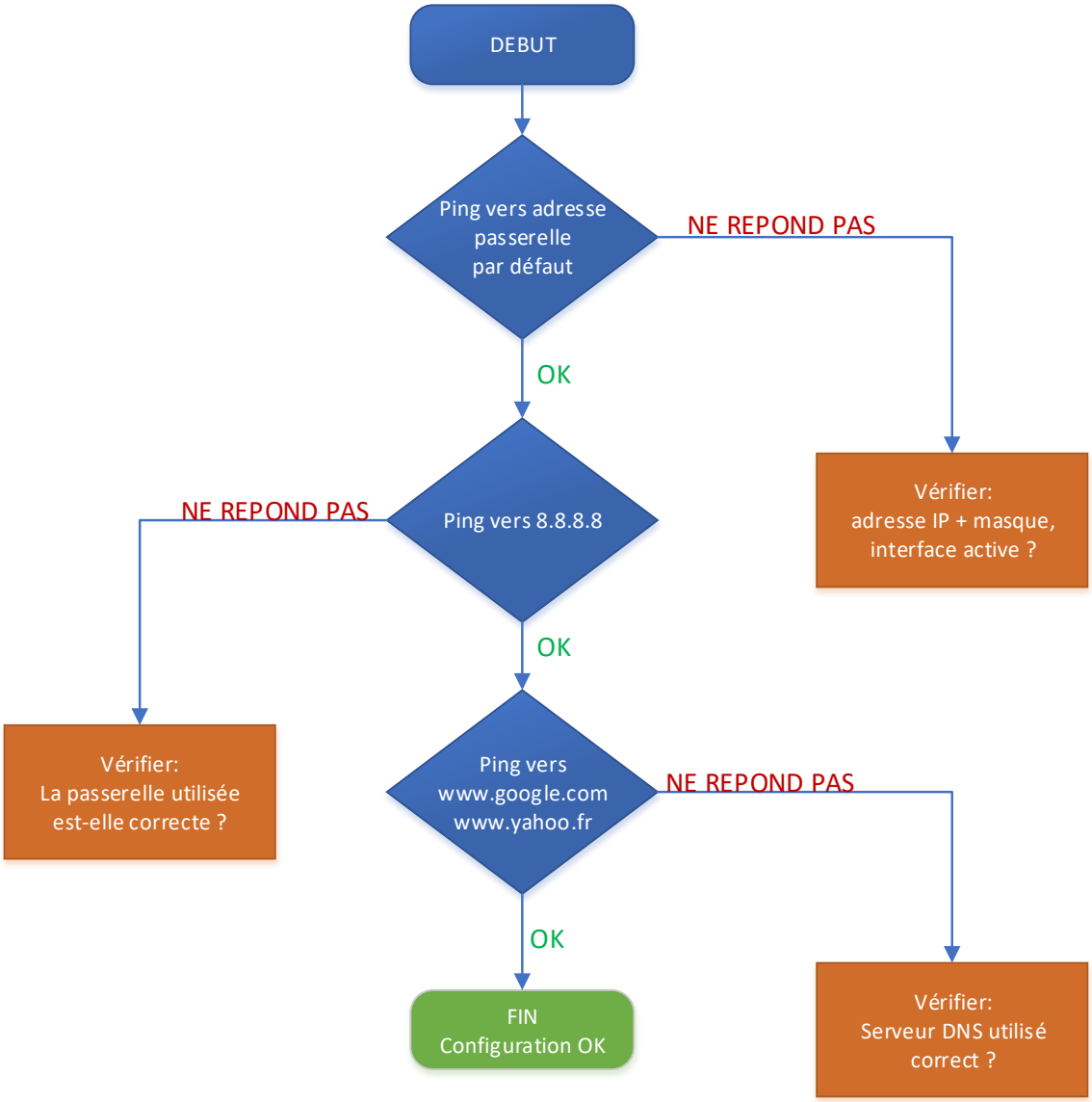

Figure 5.1 : Vérifier la configuration réseau

5.3 Quelques outils réseaux

Les outils réseaux abordés ici sont des outils courants, très souvent installés de base sur les systèmes. Ainsi, nous avons :

- **ping** qui permet de déterminer si les deux machines : locale (qui exécute la commande) et distante (qui est visée par la commande) peuvent s'échanger des messages ICMP echo-request/echo-reply. Cette commande est très souvent utilisée pour vérifier si les paramètres réseaux sont corrects.

```
$ ping 8.8.8.8
$ ping www.google.com
```

Attention ! Les firewalls sont très souvent configurés pour bloquer ces requêtes. En effet, elles ont souvent été utilisées pour des attaques DDoS.

- **traceroute** qui permet de connaître un chemin possible entre deux machines : locale (qui exécute la commande) et distante (qui est visée par la commande). Cette commande peut utiliser des paquets ICMP (option `-I`) ou UDP (option par défaut) pour tracer le chemin.

```
$ traceroute www.yahoo.fr
$ traceroute -I www.google.com
```

Comme pour la commande ping, traceroute est souvent bloqué par les firewall.

- **netstat** est un utilitaire ancien donnant des informations variées sur l'état réseau de la machine. Ainsi, en plus de donner la table de routage, il est possible de déterminer :

```
$ netstat -l -n
```

Liste les ports actuellement écoutés par un processus.

```
$ netstat -a
```

La commande affiche les connexions en cours ou en attente.

```
$ netstat -i
```

Cette commande affiche les statistiques réseaux.

- **tcpdump** est l'utilitaire qui permet de capturer le trafic réseau en ligne de commande. Cet outil est particulièrement intéressant pour analyser l'origine d'un problème (le paquet est-il envoyé ? le paquet est-il reçu ?). De nombreux paramètres de filtrage peuvent être utilisés pour cibler le trafic souhaité.

```
$ tcpdump -i eno16777736
```

Cette commande capture tout le trafic qui traverse l'interface mentionnée.

```
$ tcpdump -i eno16777736 port 25
```

Cette commande capture le trafic dont le port source ou destination est TCP/UDP 25.

```
$ tcpdump -i eno16777737 host 10.0.1.2
```

Cette commande capture le trafic dont l'adresse IP source ou destination est 10.0.1.2

Les pages de manuels donnent plus de détails sur les filtres qui peuvent être écrit.

- **wireshark** est la version graphique de l'outil **tcpdump.**

```
$ wireshark
```

- **arp** permet de gérer **la table ARP** de la machine. Pour rappel, la table ARP contient, de manière temporaire, les conversions entre les adresses MAC (ou adresses physiques) et les adresses IP. Si l'information se trouve dans la table, il n'est pas nécessaire de faire une résolution ARP[17].

Il est également possible d'insérer *des entrées statiques* dans la table grâce à cet outil (ainsi, aucune requête ne sera faite pour cette adresse IP).

[17] Envoi en *broadcast* d'une trame ethernet posant la question « Quelle machine a l'adresse IP X.X.X.X ? »

```
$ arp –a
```
Affiche le contenu de la table ARP de la machine

- **nslookup** permet d'interroger, de manière interactive, un serveur DNS déterminé. Par défaut, c'est le serveur configuré sur la machine qui est interrogé.

```
$ nslookup
> www.google.com
```
Demande la résolution du nom www.google.com

```
> server 8.8.8.8
> www.swila.be
```
Change le serveur à interroger puis demande la résolution du nom www.swila.be

```
> 193.190.64.113
```
Demande le nom correspondant à l'adresse IP 193.190.64.113 (zone DNS inverse).

5.4 Démarrage du système

Le démarrage du système est géré par une multitude de logiciels prenant le relai les uns après les autres. Ainsi, tout débute avec *GRUB* qui est très souvent installé dans les premiers secteurs du disque dur. Celui-ci permet de sélectionner le système d'exploitation à lancer (il affiche un menu avec les différents systèmes et parfois, les différentes versions du noyau Linux). Ensuite, une fois que Linux commence à démarrer, c'est *systemd* qui prend le relai. Il s'agit du mécanisme démarrant l'ensemble des services souhaités par l'utilisateur ou l'administrateur. Ainsi, c'est grâce à *systemd* que le système démarre la *configuration réseau*, le *service SSH*, ou encore l'*interface graphique*.

L'ancien gestionnaire de démarrage, nommé *System V Init* est toujours présent pour les services ne supportant pas *systemd*.

Dans la suite, nous allons aborder chaque élément pour expliquer comment

5.4.1 GRUB2

GRUB[18] est le système de chargement lancé au démarrage de Linux. Il commence par afficher le système à démarrer (cela permet à l'utilisateur de démarrer une autre version du noyau linux par exemple) et puis passe la main au système sélectionné.

Il est possible d'utiliser GRUB pour démarrer d'autres systèmes d'exploitation comme Windows (ainsi les machines des labos fonctionnent ainsi). Il est possible d'adapter le fonctionnement de GRUB en modifiant les fichiers de configuration qui se trouvent dans `/etc/grub.d`. Une fois ces fichiers modifiés, il faut régénérer le fichier de configuration au moyen de la commande :

```
$ grub2-mkconfig -o /boot/grub2/grub.cfg
```

GRUB permet également, lors du démarrage, de saisir des options particulières. Ainsi, il est possible de démarrer le système Linux *en mode single* en ajoutant simplement l'option `single`. Le mode *single* est utilisé lorsque le système ne répond plus en mode normal ou lorsque l'administrateur a

[18] Acronyme de *Grand Unified Bootloader*

perdu le mot de passe *root*. En effet, le démarrage en mode *single* ouvre un terminal en *root* sans devoir fournir le moindre mot de passe.

Pour sécuriser l'installation, il est possible d'imposer un mot de passe *grub* avant de pouvoir spécifier toute option lors du démarrage. Pour ce faire, il suffit d'ajouter, dans le fichier `/etc/grub.d/40-custom`, les lignes suivantes :

```
set superusers="root"
password root passroot
```

Nous définissons le login *superuser* à *root* et ensuite, nous fixons le mot de passe du login *root* à *passroot.* Une fois le fichier modifié, il faut régénérer la configuration via la commande `grub2-mkconfig` montrée ci-dessus.

5.4.2 SystemD

SystemD acronyme de *system daemon* est l'ensemble logiciel permettant de déterminer quels services doivent démarrer lors du lancement du système. Il est composé de nombreuses commandes. Dans le cadre de notre cours, nous nous limiterons à quelques commandes standards.

Obtenir la liste des services

```
$ systemctl list-unit-files --type service
```

Cette commande liste la liste des services activés sur le système. Si l'on souhaite voir également la liste des services non-activé, il faut utiliser l'option --all en plus :

```
$ systemctl list-unit-files --type service –all
```

Démarrer / Arrêter un service

```
$ systemctl start sshd
```

Cette commande permet de démarrer le service (ou démon) SSH.

Les arguments possibles pour la commande `systemctl` sont nombreux, les plus courants sont `start` (démarre), `stop` (arrête), `restart` (arrête et redémarre), `reload` (recharge la configuration) et `status` (affiche son état). Le dernier argument est toujours le service visé par la commande.

```
$ systemctl restart sshd
```

Cette commande permet de redémarrer le service (ou démon) SSH.

Activer / désactiver un service au démarrage

```
$ systemctl enable sshd
$ systemctl disable sshd
```

Les arguments `enable` (activer) et `disable` (désactiver) permettent de charger le service mentionné (ici, `sshd`) lors du démarrage de la machine.

5.4.3 System V Init

Comme indiqué dans l'introduction, *System V Init* est le mécanisme précédemment utilisé pour déterminer les services à démarrer lors du lancement du système. Pour des raisons de compatibilité[19], ce mécanisme est toujours supporté pour les anciens services ou ceux ne supportant pas *SystemD*. Dans la suite, nous allons donc retrouver les commandes équivalentes à celles présentées dans la section précédente mais pour les services compatibles avec *System V Init*.

Obtenir la liste des services

```
$ chkconfig
```

Cette commande liste la liste des services (compatible System V Init) présents sur le système et leurs états. C'est ainsi que l'on peut voir que Webmin et Usermin sont 2 services utilisant toujours System V Init.

Démarrer / Arrêter un service

```
$ service webmin start
```

Cette commande permet de démarrer le service (ou démon) Webmin.

Le second argument pour la commande `service` est toujours le service visé par la commande. Le dernier argument mentionne l'action souhaitée, ainsi on trouve : `start` (démarre), `stop` (arrête), `restart` (arrête et redémarre), `reload` (recharge la configuration) et `status` (affiche son état).

```
$ service webmin restart
```

Cette commande permet de redémarrer le service (ou démon) Webmin.

Activer / désactiver un service au démarrage

```
$ chkconfig --level 235 usermin off
$ chkconfig --level 235 usermin on
```

Cette commande permet de démarrer (`on`) ou arrêter (`off`) le service *usermin* lors du lancement du système et qu'il se trouve dans les niveaux[20] de démarrage `2` (mode texte multiutilisateurs, sans réseau), `3` (mode texte multiutilisateurs, avec réseau) et `5` (mode graphique, multiutilisateurs, avec réseau).

5.4.4 Démarrage en mode graphique ou mode texte

Il arrive très souvent sur des serveurs que celui-ci ne démarre qu'en mode texte. En effet, afin de ne pas encombrer la mémoire avec un environnement graphique inutile, on peut choisir de ne pas le démarrer automatiquement lors du lancement du système.

[19] CentOS 7 est la première version de la distribution CentOS supportant *SystemD.*

[20] Sans entrer dans les détails, nous considérerons toujours l'utilisation des niveaux 2, 3 et 5 dans les commandes relatives à System V Init.

```
$ systemctl isolate multi-user.target
```
Cette commande permet de « passer » dans le mode multi-utilisateur sans interface graphique.

```
$ systemctl isolate graphical.target
```
Cette commande permet de « passer » dans le mode multi-utilisateur avec interface graphique.

```
$ systemctl set-default mutli-user.target
```
Cette commande change le mode par défaut de sorte qu'au prochain démarrage de la machine, celle-ci soit en mode multi-utilisateur sans interface graphique.

5.4.5 Lancement d'une commande au démarrage

Dans bien des cas, il est intéressant de pouvoir lancer une commande lors du chargement du système. La solution idéale est, bien sûr, de créer les éléments nécessaires à l'intégration dans *SystemD*. Cependant, il est aussi possible de simplement ajouter la commande à lancer dans le fichier `/etc/rc.d/rc.local`.

Une fois la commande ajoutée, il faut rendre le fichier `/etc/rc.d/rc.local` exécutable comme suit :
```
$ chmod +x /etc/rc.d/rc.local
```

Cette opération doit seulement être faite une seule fois. Dès que le fichier `rc.local` dispose de la permission en exécution, il la garde.

5.5 Arrêt du système

Il est possible de provoquer l'arrêt ou le redémarrage du système en utilisant des commandes précises.

Pour **arrêter** le système et **éteindre** la machine :
```
$ systemctl poweroff
$ poweroff
$ halt -p
$ shutdown -h +5 "Extinction dans 5 minutes"
```

Pour **arrêter** le système :
```
$ systemctl halt
$ halt
$ shutdown -H +5 "Arret dans 5 minutes"
```

Pour **redémarrer** le système :
```
$ systemctl reboot
$ reboot
$ shutdown -r +5 "Redemarrage dans 5 minutes"
```

5.6 Exercices

5.6.1 Configuration réseau

1. En utilisant **Network Manager ou en modifiant les fichiers de configuration,** on vous demande de :
    a. Prendre note des informations réseaux reçues par DHCP
    b. Configurer statiquement l'adresse IP sur votre machine
    c. Redémarrer celle-ci
    d. Vérifier sur votre configuration est fonctionnelle en essayant de surfer sur le web

2. En utilisant **Webmin**, on vous demande :
    a. D'ajouter une interface réseau virtuelle
    b. De spécifier l'adresse IP `10.0.1.x` (ou x est votre numéro de machine)
    c. De vérifier, au moyen de la ligne de commande, que cette interface existe

5.6.2 Démarrage du système

3. A chaque démarrage, assurez-vous que l'horloge est synchronisée avec le serveur de temps `ntp1.oma.be` (voir commande et page de manuel de `ntpdate`)

4. Assurez-vous que le service *mariadb* (i.e. évolution du service de base de données MySQL) est bien démarré automatiquement au lancement du système