1

Un système de base aux petits oignons

Dans ce premier chapitre, nous allons partir d'une installation fraîche de CentOS 7 sur une machine du réseau local. La configuration post-installation de chaque serveur Linux comprend toute une série de manipulations comme la personnalisation du shell, la configuration des dépôts de paquets, l'installation d'une série d'outils, le peaufinage de la configuration réseau, etc.

L'installation de base

Le point de départ, c'est une installation minimale de CentOS 7.7 sur un PC « bac à sable » de mon réseau local. La machine – un vieux PC HP Compaq – est dotée d'un disque SSD de 60 Go, que j'ai organisé de manière relativement simple :

- une partition `/boot` de 500 Mo, formatée en `ext2` ;
- une partition `swap` de 4 Go ;
- une partition principale, formatée en `ext4`.

COMPÉTENCES Le savoir-faire acquis

Les manipulations à effectuer pour notre installation – ainsi que la configuration post-installation – ont toutes été abordées en détail dans le premier tome de cet ouvrage. Je ne reviendrai donc pas dessus. Je me contenterai tout au plus de petites piqûres de rappel par-ci par-là.

Voici les paramètres pour intégrer ma machine dans le réseau local :

- nom d'hôte : amandine.microlinux.lan
- adresse IP : 192.168.2.5
- masque de sous-réseau : 255.255.255.0
- serveur DNS : 192.168.2.1
- passerelle : 192.168.2.1

Lors de l'installation, j'ai créé un utilisateur `microlinux` avec des droits d'administrateur. Cet utilisateur fait partie du groupe `wheel` et peut invoquer des commandes avec `sudo`.

ADMINISTRATION Gérer les logs

En dehors du groupe `wheel`, j'ajoute mon utilisateur initial au groupe système `systemd-journal`, ce qui lui permet d'accéder aux logs du système sans autres privilèges :

```
$ sudo usermod -a -G systemd-journal microlinux
```

Mise à jour initiale

Une fois que le système est installé, un bon réflexe consiste à effectuer une première mise à jour.

SÉCURITÉ Mises à jour

Vous pouvez utiliser votre système tant que le distributeur met à disposition des mises à jour de sécurité afin de corriger les bogues ou les vulnérabilités potentielles. Dans le cas de CentOS 7, le système est maintenu jusqu'au 30 juin 2024.

Dans un premier temps, affichez éventuellement les mises à jour disponibles, ce qui ne nécessite pas de privilèges spécifiques :

```
$ yum check-update
...
binutils.x86_64             2.27-41.base.el7_7.1     updates
device-mapper.x86_64        7:1.02.158-2.el7_7.2     updates
device-mapper-libs.x86_64   7:1.02.158-2.el7_7.2     updates
firewalld.noarch            0.6.3-2.el7_7.2          updates
firewalld-filesystem.noarch 0.6.3-2.el7_7.2          updates
hostname.x86_64             3.13-3.el7_7.1           updates
kernel.x86_64               3.10.0-1062.4.3.el7      updates
kernel-tools.x86_64         3.10.0-1062.4.3.el7      updates
```

```
kernel-tools-libs.x86_64    3.10.0-1062.4.3.el7      updates
krb5-libs.x86_64            1.15.1-37.el7_7.2        updates
microcode_ctl.x86_64        2:2.1-53.3.el7_7         updates
polkit.x86_64               0.112-22.el7_7.1         updates
procps-ng.x86_64            3.3.10-26.el7_7.1        updates
...
```

Ensuite, mettez à jour l'intégralité du système :

```
$ sudo yum update
```

ASTUCE DeltaRPM

Certaines mises à jour ne présentent qu'une différence mineure. Dans ce cas, les DeltaRPM permettent de télécharger uniquement la différence binaire entre le paquet installé et la mise à jour, dans le but d'économiser de la bande passante. Pour utiliser DeltaRPM, il suffit de l'installer avant de procéder à l'actualisation.

```
$ sudo yum install deltarpm
$ sudo yum update
```

L'utilisation des DeltaRPM se manifestera comme ceci :

```
Delta RPMs reduced 27 M of updates to 8.7 M (67% saved)
```

Si la mise à jour a installé un nouveau noyau (paquet `kernel`), redémarrez le serveur.

Installer une panoplie d'outils

Pour l'instant, notre système est assez minimaliste et ne permet pas de travailler confortablement. La manière la plus simple de compléter notre boîte à outils consiste à installer les deux groupes de paquets `Core` et `Base`.

```
$ sudo yum group install Core
...
$ sudo yum group install Base
...
```

Le groupe `Base` fournit une série d'outils que nous sommes susceptibles d'utiliser au quotidien :

- l'éditeur `Vi` dans sa version améliorée `Vim` ;
- les outils de compression et d'archivage `bzip2`, `zip` et `unzip` ;
- le paquet `pciutils` pour gérer la configuration matérielle du serveur ;
- l'outil de téléchargement `wget` ;
- les pages `man` ;
- etc.

À partir de là, si l'invocation d'une commande retourne une erreur `commande introuvable` ou `command not found`, il suffit d'installer le paquet correspondant :

```
$ tree
-bash: tree : commande introuvable
$ sudo yum install tree
```

En français ou en anglais ?

Le poste de travail OpenSUSE sur lequel j'écris ces lignes s'affiche en français, ce qui semble normal. En revanche, lorsque je travaille en ligne de commande sur un serveur, je préfère utiliser la langue anglaise, qui constitue en quelque sorte la *lingua franca* de l'administration système sous Linux.

La configuration qui suit est optionnelle et rien ne vous empêche de garder votre système en français.

Invoquée sans options, la commande `localectl` affiche la langue du système et les différentes dispositions clavier en vigueur :

```
$ localectl
   System Locale: LANG=fr_FR.UTF-8
       VC Keymap: ch-fr
      X11 Layout: ch
     X11 Variant: fr
```

L'argument `list-locales` affiche la longue liste de toutes les locales disponibles :

```
$ localectl list-locales | grep en_US
en_US
en_US.iso88591
en_US.iso885915
en_US.utf8
```

Je décide de définir l'anglais comme langue principale de mon système :

```
$ sudo localectl set-locale LANG=en_US.utf8
```

Dorénavant, les messages de l'administrateur `root` s'affichent bien en anglais. Or, mon utilisateur `microlinux` utilise toujours le français :

```
$ echo $LANG
fr_FR.UTF-8
```

Je peux très bien basculer à la volée vers l'anglais, en redéfinissant la valeur de la variable LANG :

```
$ export LANG=en_US.utf8
```

Cette configuration n'est pas persistante, c'est-à-dire qu'elle expire lorsque je me déconnecte de ma session.

Pour utiliser l'anglais de manière permanente, je définis la variable LANG dans mon fichier ~/.bashrc :

```
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

...

# User specific aliases and functions
LANG="en_US.utf8"
export LANG
```

Personnaliser l'invite de commande

Dans sa configuration par défaut, l'invite de commande du shell Bash se présente comme ceci :

```
[microlinux@amandine ~]$
```

L'administrateur root aura la même chose, avec une invite dièse # à la place du dollar $:

```
[root@amandine ~]#
```

L'aspect de l'invite de commande peut être personnalisé, grâce à la variable d'environnement PS1 :

```
[microlinux@amandine ~]$ echo $PS1
[\u@\h \W]\$
```

Faites un premier test et remplacez \W par \w :

```
[microlinux@amandine ~]$ PS1="[\u@\h \w]\$ "
```

Cela ne change apparemment rien. Pourtant, essayez ceci :

```
[microlinux@amandine ~]$ cd /etc/yum.repos.d/
[microlinux@amandine /etc/yum.repos.d]$ cd /var/cache/yum/
[microlinux@amandine /var/cache/yum]$ cd
[microlinux@amandine ~]$
```

Désormais, l'invite de commande nous affiche le chemin complet vers le répertoire en cours.

Vous aurez probablement deviné la syntaxe pour le contenu de l'invite de commande :

- `\u` désigne l'utilisateur ;
- `\h` c'est l'hôte, c'est-à-dire la machine ;
- `\W` affiche le répertoire courant ;
- `\w` affiche le chemin complet vers le répertoire courant.

Là encore, cette personnalisation n'est pas persistante. Or, nous venons de voir que, lorsque vous ouvrez une session dans le shell Bash, celui-ci lit le contenu du fichier `~/.bashrc` pour prendre en compte des personnalisations éventuelles.

Éditez votre fichier `~/.bashrc` et enregistrez la personnalisation de votre invite de commande :

```
# User specific aliases and functions
LANG="en_US.utf8"
export LANG
PS1="[\u@\h \w]\$ "
```

Quittez la session, reconnectez-vous et naviguez un peu dans l'arborescence du système. Vous constatez que la personnalisation de l'invite de commande est désormais permanente.

Avec un peu de couleur, c'est encore mieux

Nous pouvons aller plus loin et améliorer l'aspect de notre invite de commande, en ajoutant des couleurs, ce qui non seulement est joli à voir, mais présente également un aspect fonctionnel, dans la mesure où l'invite des utilisateurs « communs mortels » se distingue clairement de celle de `root`.

Reprenez le fichier `~/.bashrc` et modifiez-le comme suit. Vérifiez bien les détails comme les guillemets simples et doubles, les caractères d'échappement, les crochets ouvrants et fermants. Et gare aux fautes de frappe ! Notez en passant que nous ajoutons quelques commentaires pour plus de lisibilité :

```
# LANG
LANG="en_US.utf8"
export LANG

# PS1
VERT='\[\033[0;32m\]'
BLANC='\[\033[1;37m\]'
GRIS='\[\033[0;m\]'
PS1="$VERT[$BLANC\u$GRIS@$BLANC\h$GRIS:$BLANC\W$VERT] \$ $GRIS"
```

Prenez en compte les modifications sans vous déconnecter de votre session :

```
[microlinux@amandine ~]$ source ~/.bashrc
```

L'invite de commande apparaît désormais en couleurs, ce qui mérite une petite explication. Les suites de caractères du genre `\[\033[1;37m\]` sont ce que l'on appelle communément des « caractères de contrôle ». Si je les ai placés dans une série de trois variables `VERT`, `BLANC` et `GRIS`, c'est uniquement pour améliorer la lisibilité de l'ensemble. Autrement, vous vous seriez retrouvé avec une définition de variable `PS1` qui ressemble à ce qui s'affiche à l'écran lorsque votre chat fait la sieste sur le clavier :

```
[microlinux@amandine:~] $ echo $PS1
\[\033[0;32m\][\[\033[1;37m\]\u\[\033[0;m\]@\[\033[1;37m\]\h\[\033[0;m\]:\[\033[1;37
m\]\w\[\033[0;32m\]] $ \[\033[0;m\]
```

Passons maintenant à l'invite de `root`, qui sera légèrement différente. Connectez-vous en tant que `root` ou exécutez simplement la commande suivante :

```
[microlinux@amandine:~] $ su -
```

Voici à quoi ressemble le fichier `/root/.bashrc` dans la configuration par défaut :

```
# .bashrc

# User specific aliases and functions
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi
```

Éditez l'invite de `root`, qui sera légèrement différente. Ajoutez la configuration suivante à la fin du fichier `/root/.bashrc` :

```
# PS1
ROUGE='\[\033[0;33m\]'
BLANC='\[\033[1;37m\]'
GRIS='\[\033[0m\]'
PS1="$ROUGE[$BLANC\u$GRIS@$BLANC\h$GRIS:$BLANC\W$ROUGE] #$GRIS "
```

Là aussi, prenez en compte la nouvelle configuration de l'invite :

```
[root@amandine ~]# source ~/.bashrc
```

Quelques alias pratiques pour la console

Maintenant que nous avons défini des invites personnalisées pour `root` et le premier utilisateur « commun mortel » du système, nous pouvons aller plus loin dans la personnalisation de la console et définir une série d'alias pratiques au quotidien. Là encore, il ne s'agit que d'une suggestion basée sur ma configuration personnelle. N'hésitez pas à expérimenter un peu pour vous l'approprier.

Redevenez un utilisateur simple et ajoutez les définitions d'alias suivantes au fichier `~/.bashrc` :

```
# User specific aliases and functions
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'
alias ll='ls -al'
alias ..='cd ..'
alias ...='cd ../..'
alias vi='vim'
```

Là encore, il faudra soit se déconnecter et se reconnecter, soit avoir recours à la commande `source ~/.bashrc` pour prendre en compte le contenu de ce fichier.

À présent, il ne vous reste plus qu'à vous connecter en tant que `root` et à définir les mêmes alias en éditant le fichier `/root/.bashrc`. Dans ce cas, vous pourrez éditer la poignée d'alias prédéfinis en les complétant.

Définir Vim comme l'éditeur principal

Certaines commandes ou applications comme `visudo`, `crontab`, `git` ou `svn` se servent des variables `EDITOR` et `VISUAL` pour lancer un éditeur de texte en mode interactif. Même si vous n'avez pas (encore) eu l'occasion de les utiliser, c'est toujours une bonne idée de renseigner ces deux variables. Concrètement, vous pourrez éditer les fichiers `~/.bashrc` respectifs de `root` et de votre utilisateur en ajoutant la section suivante :

```
# Vim
EDITOR=vim
VISUAL=$EDITOR
export EDITOR VISUAL
```

La dernière ligne fait en sorte que les variables `EDITOR` et `VISUAL` soient disponibles pour d'autres applications.

Peaufiner la configuration réseau

Supprimer NetworkManager

Dans la configuration par défaut, le réseau est géré par NetworkManager, un utilitaire Red Hat qui a pour but de simplifier l'utilisation des réseaux sous Linux, notamment sur les ordinateurs portables. En revanche, il ne sert pas à grand-chose sur un serveur et, contrairement à ce qui se dit dans les blogs un peu partout sur le Web, NetworkManager n'est **pas** nécessaire pour la gestion du réseau.

Peut-être bien que son utilisation deviendra obligatoire dans les futures versions de Red Hat Enterprise Linux et de CentOS. En attendant, c'est juste une couche d'abstraction et de complexité supplémentaire, dont on peut aisément se passer.

```
$ sudo systemctl stop NetworkManager
$ sudo yum remove NetworkManager-libnm
```

ASTUCE Pourquoi supprimer NetworkManager-libnm ?

NetworkManager est constitué de plusieurs paquets :

- `NetworkManager`
- `NetworkManager-tui`
- `NetworkManager-team`
- `NetworkManager-wifi`
- etc.

Tous ces paquets dépendent de la bibliothèque `NetworkManager-libnm`. Il suffit donc de supprimer cette dépendance commune pour se débarrasser de tous les paquets relatifs à NetworkManager.

Je vérifie si le ménage a été fait proprement et je redémarre mon serveur :

```
$ rpm -qa | grep -i networkmanager
$ sudo reboot
```

Ma configuration réseau semble toujours intacte :

```
$ ip -4 -o address show enp63s0
3: enp63s0    inet 192.168.2.5/24 brd 192.168.2.255 scope global dynamic enp63s0\
valid_lft 86356sec preferred_lft 86356sec
```

Désactiver l'IPv6

L'IPv6 est le protocole réseau du futur et nous l'utiliserons donc dans le futur. En attendant, nous pouvons sereinement le désactiver s'il ne nous sert à rien.

Dans l'état actuel des choses, les deux protocoles sont activés. Les lignes `inet` et `inet6` concernent respectivement l'IPv4 et l'IPv6 :

```
$ ip -o address show enp63s0
3: enp63s0   inet 192.168.2.5/24 brd 192.168.2.255 scope global dynamic enp63s0\
valid_lft 85944sec preferred_lft 85944sec
3: enp63s0    inet6 fe80::219:bbff:fee3:31ce/64 scope link \
valid_lft forever preferred_lft forever
```

Éditez un fichier `/etc/sysctl.d/disable-ipv6.conf` avec les paramètres suivants :

```
# /etc/sysctl.d/disable-ipv6.conf
#
# Désactiver l'IPv6
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
```

Syntaxe

Le nom du fichier n'a aucune incidence ici, du moment qu'il se termine par `*.conf`. Rien ne vous empêche de le nommer `yatahongaga.conf` en théorie. Cependant, autant lui donner un nom parlant.

Si vous ne voulez pas attendre le prochain redémarrage, prenez immédiatement en compte la nouvelle configuration :

```
$ sudo sysctl -p --load /etc/sysctl.d/disable-ipv6.conf
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
```

Dorénavant, nous n'utilisons que l'IPv4 :

```
$ ip -o address show enp63s0
3: enp63s0   inet 192.168.2.5/24 brd 192.168.2.255 scope global dynamic enp63s0\
valid_lft 85166sec preferred_lft 85166sec
```

Configuration statique

Pour l'instant, la configuration réseau de ma machine est gérée par le serveur DHCP de mon réseau local, en l'occurrence Dnsmasq, que nous aurons l'occasion de voir un peu plus loin dans cet ouvrage. Je vais remplacer la configuration réseau dynamique par une configuration statique, plus adaptée à un serveur.

L'adresse IP et le masque de sous-réseau figurent dans le fichier correspondant à mon interface réseau, en l'occurrence `ifcfg-enp63s0`. J'édite ce fichier en gardant juste les directives nécessaires :

```
# /etc/sysconfig/network-scripts/ifcfg-enp63s0
DEVICE=enp63s0
TYPE=Ethernet
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.2.5
NETMASK=255.255.255.0
```

L'adresse IP de la passerelle est renseignée dans `/etc/sysconfig/network` :

```
# /etc/sysconfig/network
GATEWAY=192.168.2.1
```

L'adresse IP de mon serveur DNS ainsi que le nom de mon domaine local se trouvent dans `/etc/resolv.conf` :

```
# /etc/resolv.conf
search microlinux.lan
nameserver 192.168.2.1
```

Le nom d'hôte simple – c'est-à-dire sans la partie domaine – est indiqué dans le fichier `/etc/hostname` :

```
amandine
```

Attention !

Ne mettez aucun commentaire dans le fichier `/etc/hostname`, sous peine de voir apparaître toute une série de bizarreries au démarrage de votre serveur. Ce fichier doit contenir le nom d'hôte simple de votre machine et rien d'autre.

Quant au fichier `/etc/hosts`, il devra ressembler à ce qui suit :

```
# /etc/hosts
127.0.0.1   localhost.localdomain localhost
192.168.2.5 amandine.microlinux.lan amandine
```

Ici, j'ai supprimé la ligne commençant par `::1`, qui constitue la référence à l'hôte local en IPv6.

Redémarrez le serveur et vérifiez la configuration réseau en envoyant une série de `ping` successifs :

- sur une adresse IP locale (comme 192.168.2.1) ;
- sur une adresse IP publique (comme 172.217.19.227) ;
- sur une machine publique via le nom d'hôte (comme google.fr).

Dans le cas où votre machine comporte plusieurs cartes réseau, vous pouvez éventuellement éditer la configuration des interfaces réseau correspondantes. Sur mon serveur de test, je dispose d'une interface `enp7s4` que je n'utilise pas. J'édite donc le fichier `/etc/sysconfig/network-scripts/ifcfg-enp7s4` comme ceci :

```
# /etc/sysconfig/network-scripts/ifcfg-enp7s4
DEVICE=enp7s4
ONBOOT=no
```

Configurer les dépôts pour Yum

Les dépôts officiels de la distribution

Les dépôts de paquets officiels sont déjà préconfigurés et utilisables tels quels. Je vais installer le *plug-in* Yum-Priorities pour les utiliser de manière prioritaire :

```
$ sudo yum install yum-plugin-priorities
```

Partant de là, j'édite `/etc/yum.repos.d/CentOS-Base.repo` en définissant une priorité maximale pour les dépôts `[base]`, `[updates]` et `[extras]` :

```
# /etc/yum.repos.d/CentOS-Base.repo
[base]
enabled=1
priority=1
name=CentOS-$releasever - Base
...
[updates]
enabled=1
priority=1
name=CentOS-$releasever - Updates
...
[extras]
enabled=1
priority=1
name=CentOS-$releasever - Extras
...
```

Je laisse le dépôt `[centosplus]` désactivé :

```
[centosplus]
enabled=0
name=CentOS-$releasever - Plus
```

Le dépôt CR

Le dépôt CR *(Continuous Release)* fournit les dernières mises à jour pour migrer en douceur vers la prochaine version mineure de CentOS avant la sortie de l'ISO officielle. La commande `yum-config-manager` fournie par le paquet `yum-utils` sert à activer ce dépôt :

```
$ rpm -q yum-utils
yum-utils-1.1.31-50.el7.noarch
$ sudo yum-config-manager --enable cr
```

Éditez le fichier `/etc/yum.repos.d/CentOS-CR.repo` en définissant la même priorité que pour les dépôts officiels :

```
# /etc/yum.repos.d/CentOS-CR.repo
[cr]
enabled=1
priority=1
name=CentOS-$releasever - cr
```

Le dépôt EPEL

Le dépôt tiers EPEL *(Extra Packages for Enterprise Linux)* fournit des paquets qui ne sont pas inclus dans la distribution CentOS. Une fois que le dépôt `[extras]` est configuré, EPEL se configure très simplement à l'aide du paquet correspondant :

```
$ sudo yum install epel-release
```

Le paquet a installé deux fichiers `epel.repo` et `epel-testing.repo` dans le répertoire `/etc/yum.repos.d`. Il suffit d'éditer le premier pour définir les priorités du dépôt `[epel]`, étant donné que les dépôts de test sont désactivés par défaut :

```
/etc/yum.repos.d/epel.repo
[epel]
enabled=1
priority=10
name=Extra Packages for Enterprise Linux 7 - $basearch

[epel-debuginfo]
enabled=0
name=Extra Packages for Enterprise Linux 7 - $basearch - Debug

[epel-source]
enabled=0
name=Extra Packages for Enterprise Linux 7 - $basearch - Source
...
```

À partir de là, nous pouvons vérifier si la gestion des priorités fonctionne comme prévu :

```
$ yum check-update
Loaded plugins: fastestmirror, langpacks, priorities
Loading mirror speeds from cached hostfile
epel/x86_64/metalink                        |  24 kB  00:00:00
 * base: fr2.rpmfind.net
 * epel: fr2.rpmfind.net
 * extras: mirrors.atosworldline.com
 * updates: mirrors.atosworldline.com
epel                                        | 5.3 kB  00:00:00
(1/3): epel/x86_64/updateinfo               | 994 kB  00:00:01
(2/3): epel/x86_64/group_gz                 |  88 kB  00:00:01
(3/3): epel/x86_64/primary_db               | 6.8 MB  00:00:01
161 packages excluded due to repository priority protections
```

Configurer l'affichage de la console

Dans la configuration par défaut, les messages de démarrage du système sont occultés par une barre de progression horizontale qui défile en bas de l'écran. Ce mode d'affichage graphique est initié par le paramètre `rhgb` *(Red Hat Graphical Boot)* :

```
# /etc/default/grub
GRUB_TIMEOUT=5
...
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="rhgb quiet"
GRUB_DISABLE_RECOVERY="true"
```

Si vous travaillez directement sur le serveur – c'est-à-dire si vous n'êtes pas connecté à la machine via une session distante dans un terminal graphique – il y a de fortes chances pour que vous trouviez la police d'affichage de la console un peu trop petite.

Rendons l'affichage plus lisible en modifiant la résolution de la console :

```
# /etc/default/grub
GRUB_TIMEOUT=5
...
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="nomodeset quiet vga=791"
GRUB_DISABLE_RECOVERY="true"
```

Prenez en compte la nouvelle configuration de Grub :

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

Sur un système UEFI, remplacez la dernière commande par celle-ci :

```
$ sudo grub2-mkconfig -o /boot/efi/EFI/centos/grub.cfg
```

Redémarrez et admirez les messages de démarrage du système, faute de pouvoir les lire à la vitesse à laquelle ils défilent.

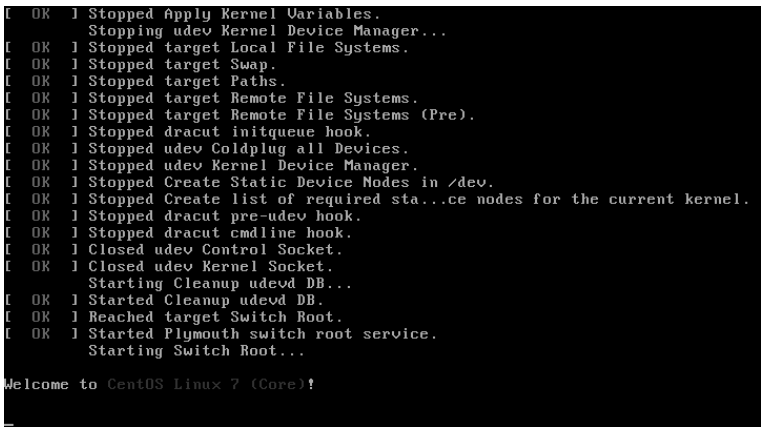

Figure 1–1
Les messages de démarrage du système.

ERREUR Régler un problème avec RPCbind

Lors du redémarrage de votre serveur, vous aurez probablement remarqué un message en rouge au beau milieu d'un joli défilé de `[ OK ]` en vert. Si le message est passé trop rapidement, nous pouvons toujours en avoir le cœur net :

```
$ journalctl -p err
-- Logs begin at Mon 2019-07-22 14:08:55 CEST, end at Mon 2019-07-22
14:20:21 CEST. --
Jul 22 14:09:00 amandine systemd[1]: Failed to listen on RPCbind
Server Activation Socket.
```

Sans rentrer dans les détails, cela tient au fait que nous avons désactivé l'IPv6. Pour remettre les choses en ordre, il suffit de forcer la reconstruction du disque mémoire initial :

```
$ sudo dracut -f -v
```

Redémarrez et vérifiez :

```
$ journalctl -p err
-- No entries --
```

Rendre sudo plus confortable

Dans le premier tome de cet ouvrage, nous avons vu qu'il existait *grosso modo* deux écoles pour l'administration d'un serveur Linux :

- ceux qui travaillent en tant que `root` ;
- ceux qui se connectent en tant qu'utilisateur normal et qui invoquent `sudo` lorsque cela est strictement nécessaire.

Les deux approches sont valables. L'avantage de `sudo`, c'est que les opérations effectuées laissent une trace dans `/var/log/secure`. En contrepartie, vous êtes peut-être las d'invoquer votre mot de passe toutes les cinq minutes. Dans ce cas, il existe une solution fort pratique.

La configuration de `sudo` s'effectue dans le fichier `/etc/sudoers`. La particularité de ce fichier, c'est qu'il ne doit **pas** être édité avec n'importe quel éditeur de texte. C'est la commande `visudo` qui doit être utilisée :

```
$ sudo visudo
```

Ajoutez ceci à la fin du fichier :

```
# /etc/sudoers
...
# Timeout
Defaults timestamp_timeout=-1
```

À partir de là, il vous suffira de saisir une seule fois votre mot de passe pour invoquer une commande avec `sudo`. Le mot de passe sera gardé en mémoire pour toute la durée de la session.

Surveiller l'état du système en un coup d'œil avec Glances

Glances est un logiciel de supervision système en ligne de commande qui vous informe en un coup d'œil de ce qui se passe sur la machine. Cet outil fort pratique a été développé par l'informaticien français Nicolas Hennion. Je m'en sers régulièrement depuis des années et je ne peux que vous le recommander. Il est disponible pour CentOS sous forme de paquet dans le dépôt EPEL.

```
$ sudo yum install glances
```

Glances offre une multitude de possibilités. Il peut fonctionner en mode *standalone* ou superviser des machines distantes, il intègre non seulement une interface en mode console, mais également un serveur Web, etc.

URL Glances

- https://nicolargo.github.io/glances/

Nous allons laisser de côté les fonctionnalités avancées de Glances pour nous contenter d'une utilisation simple.

```
$ glances
```

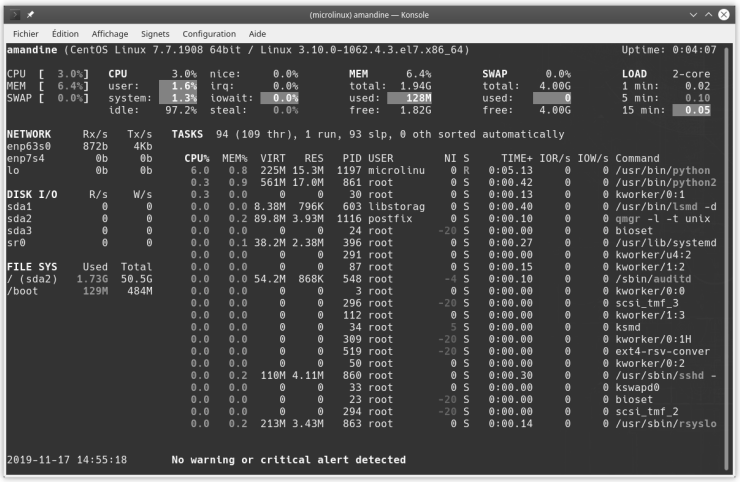

Figure 1–2
La console de Glances.

Repérez les infos système dans la console de Glances :

- le nom de l'hôte ;
- le système d'exploitation ;
- l'`uptime` ;
- la charge du processeur (`CPU`) ;
- la charge de la mémoire (`MEM`) ;
- l'utilisation éventuelle de la partition d'échange (`SWAP`) ;
- le débit des interfaces réseau ;
- le débit d'entrée/sortie des disques ;
- l'état du RAID si vous en avez un ;
- l'utilisation de l'espace disque ;
- la liste des processus.

Dans le premier tome de cet ouvrage, nous avons passé en revue toute une série de commandes qui indiquent ce qui se passe sur le système : `top`, `ps`, `df`, `cat /proc/mdstat`, etc. L'avantage de Glances par rapport à cette collection d'outils, c'est qu'il répond rapidement à la question « Qu'est-ce qui se passe sur mon serveur ? »

Les anomalies, comme les charges pathologiques de CPU et/ou de RAM ou la saturation d'une partition voire du disque entier, sont détectées immédiatement. En temps normal, les infos affichées par Glances apparaissent en vert. En cas de problème, la couleur vire progressivement au bleu, puis au violet, puis au rouge. Si vous voyez du rouge partout, vous pouvez

faire confiance à la persistante et sournoise impression que quelque chose ne tourne pas rond dans votre machine.

POUR ALLER PLUS LOIN Un article détaillé sur Glances

Si vous souhaitez découvrir les fonctionnalités avancées de Glances, lisez l'article rédigé par Nicolas Hennion et publié dans Linux Magazine France. Il est disponible en ligne gratuitement.

- https://connect.ed-diamond.com/GNU-Linux-Magazine/GLMF-182/A-la-decouverte-de-Glances-2.4

Un premier audit de sécurité avec Lynis

Lynis est un outil de sécurité développé par la société CISOfy, qui réalise un audit simple, rapide et complet d'un système Linux, Unix ou BSD. CISOfy propose une version libre et une version entreprise de l'outil. La version libre est disponible sous licence GPLv3 et ne laisse rien à désirer en termes de fonctionnalités.

URL Lynis

- https://cisofy.com/lynis/

Lynis est certes fourni par le dépôt communautaire EPEL, mais la version proposée est un peu à la traîne par rapport à la dernière version disponible chez CISOfy. Or, pour un outil d'audit de sécurité, il vaut mieux disposer de la toute dernière version. Heureusement pour nous, CISOfy met à disposition son propre dépôt de paquets pour RHEL et CentOS, qui fournit le seul paquet `lynis`.

1. Rendez-vous sur la page https://packages.cisofy.com.
2. Suivez le lien *Community Repository*.
3. Cliquez sur *Red Hat Enterprise Linux (RHEL)*.
4. Copiez-collez le fichier `/etc/yum.repos.d/cisofy-lynis.repo` sur votre machine.

Alternativement, éditez directement un fichier `/etc/yum.repos.d/lynis.repo` comme ceci :

```
# /etc/yum.repos.d/lynis.repo
[lynis]
enabled=1
priority=5
name=CISOfy Software - Lynis package
baseurl=https://packages.cisofy.com/community/lynis/rpm/
gpgkey=https://packages.cisofy.com/keys/cisofy-software-rpms-public.key
gpgcheck=1
```

Installez le paquet `lynis` et vérifiez s'il provient bien du dépôt du même nom :

```
$ sudo yum install lynis
...
---> Package lynis.noarch 0:2.7.5-100 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================
 Package      Arch       Version        Repository        Size
================================================================
Installing:
 lynis        noarch     2.7.5-100      lynis            282 k

Transaction Summary
================================================================
Install  1 Package

Total download size: 282 k
Installed size: 1.4 M
Is this ok [y/d/N]: y
```

REMARQUE Pourquoi une priorité de 5 ?

Étant donné que j'ai défini une priorité de 10 pour le dépôt EPEL, j'attribue une priorité de 5 pour celui de CISOfy. De cette manière, le paquet en provenance de l'éditeur de Lynis sera toujours prioritaire.

Invoqué sans arguments, `lynis` affiche un résumé des commandes disponibles :

```
$ lynis
```

Pour en savoir plus, on peut également consulter la page man `lynis(8)`.

Figure 1–3
Les commandes de Lynis.

L'audit du système fonctionne sans configuration préalable. On peut certes lancer un *scan* en mode non privilégié, mais il vaut mieux invoquer la commande avec les droits du superutilisateur :

```
$ sudo lynis audit system
```

L'audit complet du système est assez rapide. Au terme de l'analyse, Lynis affiche les résultats directement à l'écran, sous forme d'avertissements (`WARNING`) et de suggestions (`SUGGESTION`).

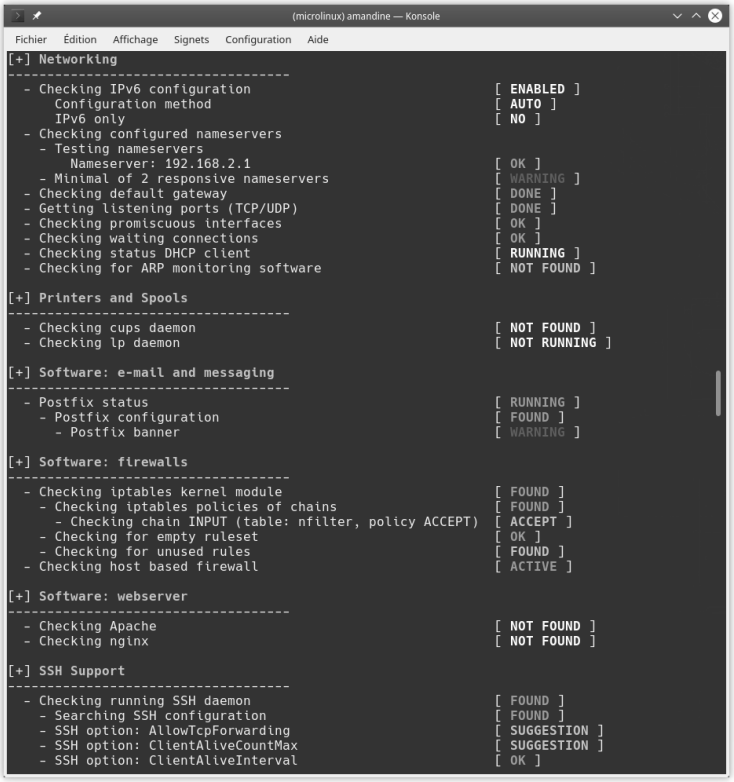

Figure 1–4
Un audit de sécurité raisonnablement complet de mon serveur.

Parallèlement, l'audit est enregistré dans le fichier `/var/log/lynis.log`. On pourra effectuer un filtrage sur les termes `Warning` et `Suggestion` :

```
$ sudo grep Warning /var/log/lynis.log
2019-07-23 15:55:39 Warning: Couldn't find 2 responsive nameservers [test:NETW-2705]
[details:-] [solution:-]
2019-07-23 15:55:40 Warning: Found some information disclosure in SMTP banner (OS or
software name) [test:MAIL-8818] [details:-] [solution:-]
```

SÉCURITÉ **Réagir face aux avertissements de Lynis**

Lynis, c'est un peu l'inspecteur de l'hygiène pointilleux qui vient faire un compte-rendu méticuleux et complet de l'état de votre cuisine. Le moindre petit détail susceptible d'être amélioré est relevé et épinglé.
Avant de vous affoler face à un tsunami d'avertissements et de suggestions, rassurez-vous. Dans le cas présent, j'ai en tout et pour tout deux avertissements.

1. Le premier relève le fait que je n'utilise qu'un seul serveur DNS dans mon réseau local.
2. L'autre concerne le serveur Postfix de ma machine, dont la configuration par défaut dévoile la version de l'application dans la bannière, ce qui pourrait éventuellement fournir quelques maigres informations à des cyberméchants résolus à prendre le contrôle de ma machine.

Ce n'est rien de bien grave dans les deux cas.

SCRIPT **CentOS « aux petits oignons » en une seule commande**

Tout au long de ce chapitre, nous avons vu en détail la configuration post-installation d'un serveur CentOS : peaufinage du shell, configuration des dépôts de paquets officiels et tiers, installation des paquets supplémentaires pour compléter la boîte à outils, etc.
Dans mon quotidien professionnel, j'utilise un script shell pour automatiser toutes ces tâches. Nous aurons bientôt l'occasion de nous initier aux scripts shell. En attendant, rien ne vous empêche d'utiliser le script `centos-7-setup.sh` que j'ai publié sur mon blog technique.

- https://www.microlinux.fr/centos-7-setup/