

Un petit nom pour votre conteneur, debriefing

I - Manuel

La première question repose sur l'utilisation du manuel. Pour rappel :

- `man -k <motif>` liste les pages du manuel dont la description courte contient `<motif>`. L'option `-K` (majuscule) permet de chercher dans le contenu complet des pages de manuel.
- le manuel ne documente pas que des exécutables, il permet aussi de documenter les appels système, des formats de fichiers, des conventions, etc. Il est organisé en sections. Ces sections sont obtenables dans le manuel de la commande `man`, si vous faites un `man man`, vous obtiendrez la liste:
 1. Programmes exécutables ou commandes de l'interpréteur de commandes (shell)
 2. Appels système (fonctions fournies par le noyau)
 3. Appels de bibliothèque (fonctions fournies par les bibliothèques des programmes)
 4. Fichiers spéciaux (situés généralement dans `/dev`)
 5. Formats des fichiers et conventions. Par exemple `/etc/passwd`
 6. Jeux
 7. Divers (y compris les macropaquets et les conventions), par exemple `man(7)`, `groff(7)`
 8. Commandes de gestion du système (généralement réservées au superutilisateur)
 9. Sous-programmes du noyau [hors standard]
- `man <section> <mot>` permet de consulter le manuel de la commande/fichier/concept/... `<mot>`.

Ainsi, la commande

```
$ man -k hostname
```

permet de lister toutes les pages de manuel en rapport avec le nom d'hôte (en: `hostname`).

Sur les conteneurs (à condition d'avoir installé les paquets `man-db` `manpages` `manpages-fr`), la liste se réduit à :

```
hostname (1)          - show or set the system's host name
hostname (5)          - Local hostname configuration file
hostname (7)          - hostname resolution description
hostnamectl (1)       - Control the system hostname
hosts (5)             - static table lookup for hostnames
org.freedesktop.hostname1 (5) - The D-Bus interface of systemd-hostnamed
ssh-argv0 (1)         - replaces the old ssh command-name as hostname handling
systemd-hostnamed (8) - Daemon to control system hostname from programs
systemd-hostnamed.service (8) - Daemon to control system hostname from programs
```

Dans la section 1 (commandes), deux commandes semblent correspondre à ce qu'on cherche : `hostname` et `hostnamectl`. La commande `ssh-argv0` est relative à SSH et semble donc hors sujet.

Dans la section 5 (fichiers de configuration), deux fichiers semblent correspondre à ce qu'on cherche : `hostname` et `hosts`.

Regardons le paragraphe de description de ces quatre pages de manuel.

1. Si on exécute la commande

```
$ man 1 hostname
```

on tombe sur une page de manuel en anglais (non traduite). Dans la description, on voit que cette commande permet d'afficher (en: *to display*) ou de définir (en: *to set*) le nom d'hôte de la machine :

« *Hostname is used to display the system's DNS name, and to display or set its hostname or NIS domain name.* »

2. Si on exécute la commande

```
$ man 1 hostnamectl
```

on tombe sur une page de manuel en anglais dont la description est très similaire à la précédente :

« *hostnamectl may be used to query and change the system hostname and related settings.* »

Dans le paragraphe synopsis, on voit que la commande `hostnamectl` demande un paramètre `{COMMAND}` dont la liste se trouve juste après le paragraphe de description. On voit que le paramètre `set-hostname` permet de définir le hostname du système.

3. Si on exécute la commande

```
$ man 5 hosts
```

on tombe sur une page de manuel en français dont la description explique que le rôle du fichier `/etc/hosts` est d'associer des adresse IP à des noms d'hôte :

« *Il s'agit d'un simple fichier texte qui associe des adresses IP avec des noms d'hôtes, une ligne par adresse IP.* »

Notre objectif n'est pas de nommer des machines sur le réseau (identifiées par leur adresse IP) mais de nommer notre propre machine, indépendamment de son ou ses adresses IP.

Remarque : le fichier `/etc/hosts` apparaîtra dans le TP sur les noms de domaines comme un ancêtre du DNS qui permet de définir des nom de domaine en local.

4. Si on exécute la commande

```
$ man 5 hostname
```

on apprend que le fichier `/etc/hostname` est utilisé pendant le boot pour configurer le nom de la machine locale :

« *The /etc/hostname file configures the name of the local system that is set during boot using the `sethostname(2)` system call.* »

Ainsi, on se dit qu'il est possible de changer le nom d'hôte du système en utilisant les commandes `hostname` et `hostnamectl` ou en modifiant le fichier `/etc/hostname`.

Remarque

Les plus curieux pourront regarder la page de manuel de `systemd-hostnamed.service` ou `systemd-hostnamed` en section 8 (correspondant à la gestion du système). La première chose qu'on remarque est qu'il s'agit des mêmes pages, et on peut voir qu'au niveau de l'arborescence, l'une est un lien symbolique vers l'autre :

```
$ ls -l /usr/share/man/man8/systemd-hostnamed.*
lrwxrwxrwx 1 root root 30 13 juil. 2021 /usr/share/man/man8/systemd-hostnamed.8.gz -> sys
hostnamed.service.8.gz
-rw-r--r-- 1 root root 906 13 juil. 2021 /usr/share/man/man8/systemd-hostnamed.service.8.g
```

Ensuite, on peut consulter le paragraphe de description de cette page :

« *systemd-hostnamed.service is a system service that may be used to change the system's hostname and related machine metadata from user programs. It is automatically activated on request and terminates itself when unused.* »

Il s'agit d'un service qui se met en marche lorsqu'il est sollicité pour changer le nom d'hôte puis termine. Pour le plaisir de la curiosité, on peut faire une petite expérience pour vérifier le comportement annoncé.

En temps normal, le service est inactif, et s'active lors de l'appel de `hostnamectl` :

```
$ systemctl is-active systemd-hostnamed
inactive
```

```
$ sudo hostnamectl set-hostname test
```

```
$ systemctl is-active systemd-hostnamed
active
```

Puis, quelques secondes plus tard, ce service se termine :

```
$ systemctl is-active systemd-hostnamed
inactive
```

II - Expérimentation

La seconde question est une petite expérimentation pour comprendre comment ces diverses façons de changer le nom d'hôte de la machine fonctionnent.

La première chose dont on se rend compte, c'est que le nom d'hôte indiqué dans le prompt du shell courant ne change pas, quelle que soit la méthode utilisée. En effet, il est fixé une fois pour toutes au lancement du shell et n'est pas mis à jour lorsque le nom d'hôte change.

Pour voir le nom d'hôte dans le prompt du shell, il faut lancer un nouveau shell. Le TP proposait de se déconnecter puis se reconnecter au conteneur. On aurait aussi pu simplement exécuter la commande `bash` (essayez !).

1. Si on exécute la commande

```
$ hostname <nouveaunom>
```

et qu'on lance un nouveau shell, on voit que le nom d'hôte est changé en `<nouveaunom>`, mais si on reboote la machine, on s'aperçoit que le nom d'hôte de la machine est le nom d'hôte précédent.

2. Si on exécute la commande

```
$ hostnamectl set-hostname <nouveaunom>
```

et qu'on lance un nouveau shell, on voit que le nom d'hôte est changé en `<nouveaunom>`, mais si on reboote la machine, on s'aperçoit que le nom d'hôte de la machine est encore `<nouveaunom>`.

3. Si on modifie le fichier `/etc/hostname` en lui mettant un nouveau nom et qu'on lance un nouveau shell, on voit que le nom d'hôte n'est pas modifié. Par contre, si on reboote la machine, on voit que le nom d'hôte de la machine est le nouveau nom.

Si on revient à l'étape 2, on s'aperçoit que la commande `hostnamectl` modifie le fichier `/etc/hostname`.

Remarque

Les 3 méthodes sont complémentaires et on si on lit plus en détail leurs pages de manuel, on peut voir qu'elles se citent mutuellement (paragraphes `SEE ALSO` et `FILES`).

Remarque

On peut faire une recherche sur le web pour mettre à jour le nom d'hôte dans le prompt du shell courant (pour ne pas avoir à lancer un nouveau shell), on peut par exemple chercher avec les mots clefs "refresh hostname shell prompt" (essayez !).

III - Interprétation : configuration temporaire *vs* configuration persistente

La troisième question demande d'interpréter l'expérimentation précédente.

Souvent, avec une commande vous modifiez l'état du noyau qui est en cours d'exécution, et cet état est perdu à l'extinction ou au redémarrage de votre machine. Pour modifier de façon permanente une configuration afin qu'elle soit rechargée au redémarrage, cela se fait la plupart du temps en modifiant un fichier de configuration (qui, pour rappel, se trouve dans `/etc`).

Ainsi, quand vous utilisez la commande `hostname`, vous modifiez le nom d'hôte de votre machine au niveau du noyau qui est en RAM, de sorte que quand la machine redémarre, cette information est perdue. Ainsi, au redémarrage, c'est le contenu du fichier `/etc/hostname` qui sera lu et utilisé pour le nom d'hôte. Si vous modifiez le fichier `/etc/hostname`, alors c'est le contenu de ce fichier qui sera utilisé au reboot, mais lorsque vous modifiez ce fichier, le noyau ne va pas de lui-même le consulter pour se mettre à jour du nouveau nom. La commande `hostnamectl set-hostname <nom_d_hote>` fait les 2 en même temps : elle modifie le fichier `/etc/hostname` pour les prochains démarrages ET elle avertit le noyau du changement de nom d'hôte.

Remarque importante : généralisations

Ce mécanisme est assez courant. Voici quelques exemples importants à méditer :

- Lorsque vous voulez monter un système de fichiers sur un répertoire, vous faites un

```
# mount <filesystem> <répertoire>
```

Mais au redémarrage, le système de fichiers ne va pas être monté automatiquement, car cette information n'a pas été stockée de façon persistante. Si vous voulez que ce montage se fasse automatiquement, il faut ajouter une ligne dans le fichier de config `/etc/fstab`.

1. Consultez le manuel du fichier de configuration (section 5) `fstab`.
2. Consultez le fichier `/proc/mounts`
3. Consultez le fichier `/etc/fstab` de votre système

- Pour configurer le réseau, vous utilisez la commande `ip` (et éventuellement `dhclient`, `wpa_supplicant`, etc), mais si vous voulez que la configuration soit utilisée aux prochains démarrages, il faut modifier un ou plusieurs fichiers dans le répertoire `/etc/systemd/network/` (note : anciennement dans les systèmes Debian, la configuration persistante du réseau se trouvait dans le fichier `/etc/network/interfaces`, avec une syntaxe pas toujours univoque).
- Pour démarrer un service, vous utilisez la commande

```
# systemctl start <service>
```

Mais si vous voulez que le service soit démarré au reboot, vous faites plutôt :

```
# systemctl enable <service>
```

Vous me direz que là il s'agit de 2 commandes et pas une commande *vs* un fichier de configuration. OK c'est pas faux. Sauf que, si vous faites une expérience à base de `find -newer`, vous vous rendrez compte que `systemctl enable <service>` modifie le répertoire `/etc/systemd/system`. Vous pouvez par exemple voir qu'un lien symbolique `/etc/systemd/system/multi-user.target.wants/nginx.service` est ajouté ou supprimé lorsque vous faites `systemctl enable nginx` ou `systemctl disable nginx`.

1. Faites l'expérience !

Remarque : on avait pas dit que « tout est fichier » ?

Si si. Comme on l'a vu en cours, si vous faites un `man 5 proc`, vous apprendrez que « *The proc filesystem is a pseudo-filesystem which provides an interface to kernel data structures. It is commonly mounted at*

/proc. »

Si, dans la lecture de `man 5 proc` vous recherchez la chaîne `hostname` (en tapant `/hostname`), vous apprendrez l'existence du fichier `/proc/sys/kernel/hostname` dont le contenu est le `hostname` courant du système, et qui est modifiable.

Ainsi, la commande

```
$ cat /proc/sys/kernel/hostname
```

est équivalente à la commande

```
$ hostname
```

et la commande

```
# echo 'darkstar' > /proc/sys/kernel/hostname
```

est équivalente à la commande

```
# hostname 'darkstar'
```

La différence entre le fichier `/etc/hostname` et le fichier `/proc/sys/kernel/hostname` est que le fichier `/etc/hostname` fait partie d'un système de fichier "classique" installé dans un block device correspondant à une mémoire de masse (un disque dur, une clef USB, etc) qui ne s'efface pas lorsqu'on éteint la machine.

Le fichier `/proc/sys/kernel/hostname` fait partie du système de fichier `procfs` qui est une interface permettant d'accéder à certaines informations du noyau et des processus. Ces informations sont stockées en RAM, et ne survivent pas à un redémarrage du système.

IV - À propos de l'IETF et des RFC

En feuilletant la page de manuel de `hostname` en section 7 ("divers", dont les conventions), il est fait mention dans le paragraphe `SEE ALSO` des RFC 1123 et 1178.

La plupart des normes (protocoles) qui régissent le fonctionnement d'internet sont rédigées par l'IETF (Internet Engineering Task Force), qui est un groupe informel (tout le monde peut y participer). Les normes prennent la forme de textes appelés RFC, pour *Request For Comments* (fr: demande de commentaires).

Voici quelques exemples de RFC concernant des protocoles rencontrés cette année :

- IPv6 : RFC 2460 (1998), RFC 8200 (2017)
- TCP : RFC 793
- HTTP : RFC 1945 (1996), ..., RFC 7540 (2015)
- SSH : RFC 4250 à 4254 (1995)
- transmission de paquets par pigeon voyageur : RFC 1149 (1er avril 1990)

Certaines RFC sont précisées, augmentées ou corrigées par des RFC ultérieures.

Certaines RFC sont traduites en français par des amateur·es, voir les sites <http://www.rfc.fr> et <http://abcdrfc.free.fr/>

La RFC 1178 n'est pas une RFC importante, elle ne définit pas un protocole, mais donne des conseils sur la façon de choisir un nom de domaine. Elle a l'avantage de se lire facilement.

- La RFC 1178 : <http://www.ietf.org/rfc/rfc1178.txt>
- une traduction de la RFC 178 en français : <http://www.rfc.fr/rfc/fr/rfc1178.pdf>