Projet 2019-2020 : Raspberry Pi comme serveur d'accès distant

- Accès distant à votre réseau local (Raspbian, PiVPN, Pi-hole, SSH, utilitaires)
- Présentation par Raymond Ouellette
- Documentation prête, expérimentation fonctionnelle
- Expérimentations personnelles, entraide

Membres intéressés:

- À venir ; inscrivez votre nom ici ou écrivez-nous info@linuq.org)

Résumé

Pour accéder de l'extérieur à notre réseau local le nano-ordinateur Raspberry Pi est le choix tout indiqué : on peut le laisser ouvert en permanence car il est silencieux et peu énergivore. En lui ajoutant les serveurs PiVPN et Pi-hole il permet de se connecter sur son propre serveur OpenVPN, de gérer les ordinateurs de notre réseau – même de les démarrer et de les arrêter – et de filtrer en amont les publicités en tout temps. En prime, si vous êtes à l'extérieur, la connexion OpenVPN vous offrira non seulement les services d'un VPN mais aussi celui du filtre de publicité Pi-hole comme si vous étiez chez vous.

Le Raspberry Pi

Le Raspberry Pi est un nano-ordinateur monocarte à processeur ARM conçu par des professeurs du département informatique de l'université de Cambridge (Royaume-Uni) dans le cadre de la fondation Raspberry Pi [https://fr.wikipedia.org/wiki/Fondation_Raspberry_Pi].

Parfaitement silencieux et peu énergivore, il est suffisamment puissant pour agir comme serveur. Le modèle 3B+ a un processeur Broadcom BCM2837B0 64 bit à quatre cœurs ARM Cortex-A53 cadencé à 1,4 GHz, 1 Go de mémoire vive, une carte vidéo Broadcom VideoCore 1080p30, une puce WiFi Dual-band 802.11ac, la version 4.2 du Bluetooth, 4 ports USB 2, 1 port HDMI, une sortie stéréo Jack 3,5 mm (le son 5.1 sur la prise HDMI), une unité de lecture/écriture microSD, une carte réseau 10/100/1000 Ethernet (max 300Mbps) et un connecteur 40pin GPIO. Il est de la taille d'une carte de crédit et pèse 45g sans son boîtier. Son alimentation électrique de 5V est assurée par une prise Micro-USB ou avec GPIO header ou avec Power Over Ethernet. La version 4, plus puissante, est maintenant disponible.

Raspbian ou Linux Debian pour processeur ARM

La version Linux Debian pour ARM, nommée Raspbian [https://www.raspberrypi.org/downloads/raspbian/], est le système d'exploitation suggéré pour son fonctionnement. On retrouve deux versions de Raspbian : la version Desktop (autour de 1,4 Go) avec un bureau graphique complet et une foule de logiciels et la version Lite (autour de 400 Mo), parfaite comme serveur, même si l'interface graphique est absente. De toute façon, il est très facile d'ajouter ensuite les logiciels de son choix à chacune des versions (y compris l'interface graphique), c'est du Debian après tout!

Le système s'installe sur une carte de mémoire microSD d'une taille minimale suggérée de 8 Go. Il n'y a pas vraiment d'avantages à prendre une carte trop volumineuse parce que l'espace restant, qui peut bien sûr être utilisé, est d'un accès plus lent que celui offert par une simple clé sur un des ports USB du Raspberry Pi. Un disque rigide externe peut être branché à condition qu'il soit alimenté par sa propre source d'énergie.

La carte microSD sera préparée (formatage et installation) sur un ordinateur Linux en l'insérant soit dans la fente du lecteur de carte SD soit dans un port USB avec un adaptateur USB (parfois inclus à l'achat de la carte). Assurez-vous de prendre une carte **de bonne qualité** pour éviter la corruption fréquente de fichiers sur les carte microSD bon marché.

L'installation présentée ici est dite sans affichage ou **headless** en anglais : on peut certes connecter un clavier USB, une souris et un écran sur le port HDMI pour accéder au Raspberri Pi quand sa carte microSD aura été préparée puis insérée dans le petit lecteur du Pi, mais il est tout aussi possible d'accéder au Pi à distance grâce à une connexion ssh et de compléter et peaufiner son installation dans un terminal. Comme le Raspberri Pi est destiné à être un serveur, aussi bien savoir comment y accéder à distance avec le meilleur outil approprié : openssh.

Préparation de la carte microSD

La préparation se fait sur un ordinateur Linux et non sur le Raspberry Pi.

- Insérez la carte microSD (dans son adaptateur) dans la fente ou le port USB
- Ouvrez un terminal et tapez `lsblk` validée par la touche d'entrée, vous obtiendrez semblable à ceci :

```
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda      8:0    0 931,5G  0 disk
├─sda1   8:1    0    60G  0 part /
├─sda2   8:2    0 119,6G  0 part /home
├─sda3   8:3    0 571,9G  0 part /partage
└─sda4   8:4    0   150G  0 part /secur
sdc      8:48   1  29,3G  0 disk
└─sdc1   8:49   1  29,3G  0 part /run/media/usager/DE_31Go
sr0     11:0    1  1024M  0 rom
```

- La carte microSD est ici le périphérique (device) `/dev/sdc` de 29,3G, je la reconnais à sa taille. Notez bien le nom, ce sera celui utilisé pour l'installation du système d'exploitation Raspbian.
- `cd le_répertoire/de_l'image` – ou ouvrez votre explorateur de fichier dans le répertoire où

se trouve le fichier image de la version de Raspbian que vous avez déjà téléchargé, cliquez à droite et ouvrez dans un terminal
- `ls -lh` pour voir le contenu du répertoire et s'assurer que l'image est bien présente – par exemple :

```
… 352M 16 avr 01:31 2019-04-08-raspbian-stretch-lite.zip
```

- `unzip -p 2019-04-08-raspbian-stretch-lite.zip | sudo dd of=/dev/sdc bs=4M conv=fsync` pour décompresser à la volée et copier l'image sur le périphérique `/dev/sdc`.
  Attention à bien choisir le nom du périphérique qui sera écrit parce que l'opération est sans appel :
  `dd` (pour disk duplicate) `of` (pour output file) `bs` (pour block size, la taille des tampons lus et écrits pour que la copie soit assez rapide) et `conv=fsync` pour synchroniser l'écriture avec l'affichage à l'écran et en voir la progression. L'opération prend plusieurs minutes si la carte est volumineuse. Lorsque l'invite du terminal revient, tapez `sync` et attendez qu'il vous rende la main à nouveau, on est ainsi assuré que tout a été écrit sur la carte microSD.
- Quand la carte est prête, tapez `lsblk -f` et vérifiez le nom du périphérique de votre carte microSD.

```
NAME   FSTYPE LABEL   UUID FSAVAIL FSUSE% MOUNTPOINT
sda
├─sda1 ext4           b438776b-2068-406f-85ee-63e525df4d77   38,9G    29% /
├─sda2 ext4           f9a120b8-b953-4634-84b6-798420d2465a   96,4G    13%
/home
├─sda3 ext4   partage 6d651be3-0cf0-3d0c-bcb5-754444954017  255,3G    49%
/partage
└─sda4 ext4   secur   f9d2a2f0-f725-4078-b46a-32e59c62b5e8  115,1G    16%
/secur
sdc
├─sdc1 vfat   boot    1174-F769
└─sdc2 ext4   rootfs  4bab068f-f791-27a4-8f8f-b4cf59f072a1
sr0
```

- Avec un `sudo gparted /dev/sdc` vous pouvez maintenant examiner la structure de votre carte. Le plus simple sera d'étendre la seconde partition (celle qui contient tout le système) avec le logiciel `raspi-config` que nous utiliserons plus loin. Quittons gparted et démontons la carte.
- Retirez la carte du lecteur et insérez-la de nouveau pour qu'elle soit lue par le système. Elle devrait apparaître et vous remarquerez une partition nommée `boot`. Créez dans cette partition un fichier vide nommé `ssh` (sans extension). En mode graphique (dans Nautilus) cliquez à droite → Nouveau document → Nouveau pour créer le document vide. Enlevez l'extension et nommez le fichier `ssh`. Sinon cliquez à droite dans la partition ouverte avec Nautilus, → Ouvrir dans un terminal et tapez `touch ssh` et fermez le terminal.
- **Si vous avez modifié la taille de la partition rootfs à l'étape avec gparted**, il faudra aussi corriger le fichier `cmdline.txt` pour que le démarrage du système soit correctement passé à cette partition. Supposons que votre partition principale `rootfs` soit `/dev/sdc2` alors faites la commande suivante dans un terminal :
  `sudo blkid -s PARTUUID -o value /dev/sdc2`
  vous obtiendrez une réponse semblable à celle-ci :

`4d6dceb1-02`
Utilisez cette valeur dans le fichier `/boot/cmdline.txt` après `root=PARTUUID=` en remplaçant la valeur incorrecte par la nouvelle au moyen de votre éditeur de texte préféré.
- Démontez les partitions de la carte qui est maintenant prête.
- Insérez la carte microSD dans le Raspberry Pi; branchez-y le câble réseau RJ-45 qui provient du routeur et branchez l'alimentation électrique. Le Raspberry Pi va démarrer et vous verrez clignoter les 2 témoins lumineux.

Préparation du Raspberry Pi

D'abord accéder au Raspberry Pi

Votre nano-ordinateur est maintenant prêt à être configuré après son premier démarrage. Mais pour y accéder il faut connaître son adresse IP; comme aucun périphérique n'y est branché (écran, souris, clavier), seul l'accès ssh est possible. On peut découvrir l'adresse IP de plusieurs façons :

1. En interrogeant le routeur depuis notre ordinateur principal dans un fureteur; dans le cas de Vidéotron par exemple il faut utiliser l'adresse `http://192.168.0.1` et se rendre dans les réglages experts à la liste des clients du serveur DHCP. Le nom du Raspberry Pi peut apparaître; sinon en examinant les adresses on pourra déduire laquelle appartient au Pi vu que l'on connaît celles des autres appareils du réseau local.
2. `arp -a` dans un terminal va interroger la cache ipV4 du noyau et afficher les adresses IP et MAC connues. arp signifie « Address Resolution Protocol »
3. `nmap -sn 192.168.0.0/24` dans un terminal va scanner toutes les adresses du modem (192.168.0.X) qui sont actives (up), celle du Pi s'y trouve!

Tentez un premier accès ssh de la façon suivante : `ssh pi@192.168.0.150` (adresse supposée) avec *raspberry* comme mot de passe. Vous obtiendrez un résultat semblable à ceci :

```
$ ssh pi@192.168.0.150
The authenticity of host '192.168.0.150 (192.168.0.150)' can't be established.
ECDSA key fingerprint is a6:45:7d:78:4b:c8:52:72:a0:6b:52:37:c8:e6:73:45.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.150' (ECDSA) to the list of known hosts.
pi@192.168.0.150's password:
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Nov 15 15:11:27 2018
```

Vous pouvez peut-être avoir aussi ce message : `NOTICE: this Raspberry Pi has not been fully configured. Please run 'sudo raspi-config'` Qu'il apparaisse ou non, ce sera la prochaine étape.

Configuration de base du Raspberry Pi

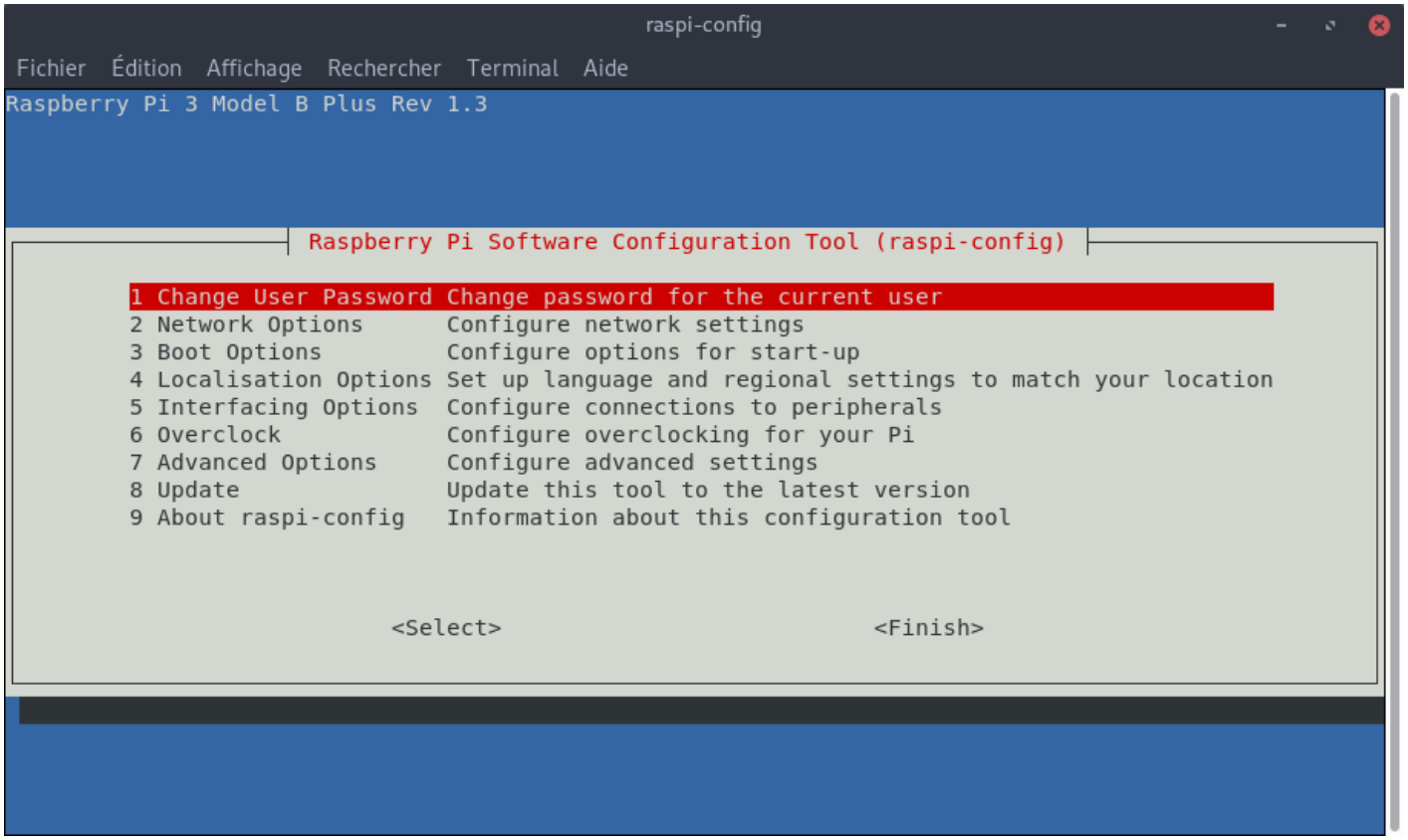
Au premier branchement on se connecte comme usager *pi* avec *raspberry* comme mot de passe, c'est ce même mot de passe que l'on donne après cette commande sudo. L'usager pi a les droits de root, il est donc primordial de changer tout de suite son mot de passe pour notre protection. Je préfère créer un autre usager à qui donner les droits d'administration. On peut ensuite supprimer le compte de pi mais plusieurs utilisateurs de Raspberry Pi le déconseille.

Pour ajouter un usager sur le Raspberry Pi

`sudo adduser usager` et suivre les directives à l'écran (évitez l'inutile, nom d'usager et mot de passe suffisent). Puis on donne les droits de sudo à l'usager avec la commande `sudo visudo`. Dans le fichier qui s'ouvre, ajoutez la ligne `usager ALL=(ALL:ALL) ALL` sous la ligne `root ALL=(ALL:ALL) ALL` de la section `# User privilege specification`. Ctrl + O pour écrire et Ctrl + X pour quitter visudo. Adaptez le nom de l'usager à vos besoins.

Configurons maintenant le Raspberry Pi (en tant qu'usager pi)

`sudo raspi-config` lance l'utilitaire de configuration du Raspberry Pi.



L'option 1 de raspi-config sert donc ici à changer le mot de passe de pi (puisqu'on s'est connecté comme usager pi).

On se déplace au clavier dans cette interface avec les flèches et la touche Tab.

L'option 2 concerne le réseau. Donnez un nom au Raspberry Pi (pas d'espace ni de signes diacritiques –

accents, cédilles, etc – privilégiez un nom assez simple et descriptif. Il apparaîtra toujours dans l'invite du terminal. Profitez-en pour activer le WIFI (nom du SSID et mot de passe). **L'option 3** concerne le mode de démarrage. Choisissez B1 Desktop/CLI pour l'interface en mode texte ou ligne de commande. C'est plus léger, plus rapide et moins gourmand en ressources. Tous les logiciels graphiques installés sur le pi se lanceront quand même et c'est le serveur Xorg de votre ordinateur qui sera utilisé pour leur affichage. Il n'y a pas d'avantage à choisir un autre mode. **L'option 4** sert à l'internationalisation du Raspberry Pi : français Canada, fuseau America Eastern (Toronto), clavier cf et CA pour le pays du WIFI. **L'option 5** concerne les accès à l'appareil. Activez P2 SSH, P4 SPI, P5 I2C, P6 Serial et P7 1-Wire, ce sont les choix conseillés. L'overclocking n'est pas nécessaire, on évitera **l'option 6**. Dans **l'option 7** seul le choix A1 est à considérer si vous n'avez pas utilisé tout l'espace disponible de la carte microSD quand vous l'avez examiné et préparé avec Gparted. Les autres options seront peu utiles en mode serveur. On ferme cette application et on redémarre le Raspberry Pi avec la commande `sudo reboot`

Mise-à-jour logicielle du Raspberry Pi

Le Raspberry Pi est redémarré et vous y êtes branché par ssh (sur le compte de pi avec son nouveau mot de passe bien sûr, ou comme nouvel usager). Il faut maintenant s'assurer, comme dans tout système Linux, que tous les logiciels sont à jour pour la sécurité et la fonctionnalité. `sudo apt update ; sudo apt dist-upgrade`

La première commande met à jour le cache des logiciels (l'ensemble des logiciels disponibles sur les différents miroirs de la distribution Raspbian), puis lorsque l'opération est complétée (le point-virgule), faire une mise à jour plus complète (dist) qui peut inclure le noyau Linux.

`sudo apt-get install software-properties-common mc` pour mieux gérer les mises à jour.

Le logiciel *unattended-upgrades* sera installé, il permet d'effectuer automatiquement des mises-à-jour de sécurité, fort utile pour un serveur sans affichage toujours branché dans l'Internet. Lorsque *unattended-upgrades* est installé, ajoutez les logiciels suivants pour le paramétrer :

```
sudo apt-get install mailutils s-nail update-notifier-common
```

Puis nous allons éditer avec le logiciel *mc* (ou avec l'éditeur de texte de votre choix) le fichier `/etc/apt/apt.conf.d/50unattended-upgrades` : `sudo mc /etc/apt/apt.conf.d` Il faudra décommenter (en enlevant les deux // du début) les lignes suivantes :

```
Unattended-Upgrade::Mail "usager";
Unattended-Upgrade::MailOnlyOnError "true";
Unattended-Upgrade::Automatic-Reboot "true";
```

Évidemment changez le nom de l'usager qui recevra le courriel!

Toujours dans le même répertoire, éditez le fichier `/etc/apt/apt.conf.d/20auto-upgrades` pour qu'il contienne ces lignes :

```
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Download-Upgradeable-Packages "1";
APT::Periodic::Unattended-Upgrade "1";
APT::Periodic::Verbose "1";
```

```
APT::Periodic::AutocleanInterval "7";
```

Vous pouvez tester si cela fonctionne avec cette commande : `sudo unattended-upgrade -d -v –dry-run`

Enfin, pour activer ces mises-à-jour automatique avec unattended-upgrade, exécutez cette dernière commande : `sudo dpkg-reconfigure –priority=low unattended-upgrades` Répondez *oui* et acceptez la ligne de configuration par défaut qui est de ne mettre à jour automatiquement que les logiciels affectés par un correctif de sécurité. Les autres mises à jour devront être faites manuellement sauf si vous décommentez les autres types de mises à jour dans le fichier `/etc/apt/apt.conf.d/50unattended-upgrades`.

Les mises à jour manuelles se font de la façon suivante : `sudo apt update ; sudo apt upgrade` Faites-les de temps en temps, 1 ou peut-être 2 fois par mois suffisent grâce aux mise-à-jour automatiques de sécurité.

Enfin, parlant de sécurité, un serveur ssh, même si on change son port habituel d'accès, demeure sensible aux attaques par force brute. Installez le logiciel *fail2ban* avec `sudo apt install fail2ban` : le logiciel analysera les connexions et bannira les tentatives d'accès non désirées.

On activera ensuite *fail2ban* avec ces trois commandes (une seule fois, la dernière rend le service permanent) :

```
sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
sudo systemctl start fail2ban.service
sudo systemctl enable fail2ban.service
```

Ajouts de logiciels pratiques

Même si votre Raspberry Pi contient déjà par défaut une pléthore de logiciels (surtout si vous avez choisi la version Desktop de Raspbian), je vous suggère les ajouts suivants :

Deux fureteurs Internet légers, le premier graphique et le second texte seulement : `sudo apt install midori elinks`

Des utilitaires : `sudo apt install screen cronie libcanberra-gtk3-module ethtool wakeonlan etherwake ntfs-3g ranger ttf-mscorefonts-installer lshw locate avahi-daemon gnome-terminal jhead exiv2 pdftk mupdf imagemagick libgcj16-dbg libgcj16-awt mupdf-tools xpdf-utils`

Et j'en oublie sûrement!

Sauvegarde de votre installation

Le point faible du Raspberry Pi est son fonctionnement à partir de la carte microSD qui peut se corrompre facilement (par exemple lors d'une panne de courant). Utiliser une carte microSD de qualité peut aider, mais l'idéal est la bonne vieille copie de sauvegarde de la carte, voire un doublon de votre installation sur une autre carte microSD.

Cette étape peut se faire après l'installation des deux autres serveurs PiVPN et Pi-hole que nous verrons plus loin.

- Éteignez le Raspberry Pi : `sudo shutdown -h now`
- Insérez (avec votre adaptateur au besoin) votre carte microSD dans un ordinateur sous Linux et faites un `lsblk -f` pour connaître le nom du périphérique de votre carte (supposons `/dev/sdd`)
- Utilisez la commande dd pour faire une copie de votre carte microSD – assurez vous d'avoir suffisamment de place pour la copie qui occupera **au minimum le même espace** que la taille de votre carte : `sudo dd bs=4M if=/dev/sdd of=raspbian_20190711_raspi.img status=progress oflag=sync`
- Réduisez ensuite la taille de l'image à l'aide du script *pishrink.sh* que vous lancerez comme administrateur. Ce script que vous trouverez à https://github.com/Drewsif/PiShrink est peut-être même déjà disponible dans les dépôts de votre distribution Linux préférée. Le script réduira considérablement la taille de votre image en supprimant la partie inutilisée de la partition `/rootfs` et rendra votre image auto-expansible à la taille maximale de votre carte microSD lors de son premier démarrage sur un Raspberry Pi. L'image de mon installation complète sur une carte de 16 Go a été réduite à 2,4 Go!
- On réinstalle une image avec la même commande dd en s'assurant toujours du bon nom de périphérique, par exemple : *`sudo dd bs=4M if=raspbian_20190711_raspi.img of=/dev/sdd status=progress oflag=sync`* Si l'image a été réduite à 2,4 Go par exemple, elle peut être copiée sur une carte aussi petite que 4 Go mais préférez une carte d'au moins 8 Go. Après l'insertion de la copie de la nouvelle carte dont l'image avait été réduite avec le script pishrink.sh dans le Raspberry Pi, la partition `rootfs` sera automatiquement ajustée à la taille maximale disponible sans aucune intervention de votre part. Vous pourrez, lorsque toute activité aura cessé, accéder au nano-ordinateur en *ssh*.

Gardons le contact!

Maintien du lien IP

Votre réseau à la maison est accessible au moyen d'une adresse IP. Sauf si vous avez acheté une adresse permanente, vous devez utiliser celle attribuée par votre fournisseur Internet. Cette adresse ne change pas souvent; elle peut rester la même pendant des mois, mais votre fournisseur Internet peut la changer sans avertissement. Voici donc un petit script qui lit l'adresse externe de votre routeur, vérifie si elle a changé et, le cas échéant, vous envoie un courriel contenant la nouvelle adresse.

```
#!/bin/bash
#Script to report public IP address change
#By: Ronny L. Bull
#The file that contains the current pubic IP
EXT_IP_FILE="/home/usager/.config/notre_ip"
#Get the current public IP from whatsmyip.com
CURRENT_IP=$(curl http://ifconfig.me/ip)
#Check file for previous IP address
if [ -f $EXT_IP_FILE ]; then
KNOWN_IP=$(cat $EXT_IP_FILE)
else
KNOWN_IP=
```

```
fi
#See if the IP has changed
if [ "$CURRENT_IP" != "$KNOWN_IP" ]; then
echo $CURRENT_IP > $EXT_IP_FILE
#If so send an alert
echo "Nouvelle adresse $CURRENT_IP" | mailx -v -s "Nouvelle adresse"
adresse@posteo.net
else
#If not just report that it stayed the same
echo "Adresse IP inchangée" > /dev/null
fi
```

Adaptez selon vos besoins. J'ai placé ce script que j'ai nommé `verif_IP` dans mon répertoire `.config` (n'oubliez pas le point devant config, utilisez `mc` pour voir le contenu du répertoire). On rend le script exécutable et on lui donne seulement des droits d'accès pour l'usager : `sudo chmod 700 ~/.config/verif_IP ; sudo chown usager:usager ~/.config/verif_IP` Adaptez à votre usager choisi (pi ou un autre que vous aurez créé). Créez le fichier qui contiendra l'adresse IP : `touch /home/usager/.config/notre_ip` Pour que vous receviez un courriel il faut créer un fichier `.mailrc` dans la racine de l'usager avec ce contenu :

```
set smtp-use-starttls
set ssl-verify=ignore
set smtp=posteo.de:587
set smtp-auth=login
set smtp-auth-user=usager@posteo.net
set smtp-auth-password="super_mot_de_passe"
set from="usager@posteo.net(ordinateur)"
```

Adaptez selon votre fournisseur de courriel, j'ai mis ici l'exemple de posteo. Ce fichier sera utilisé par le logiciel `s-nail` quand le script `verif_IP` sera exécuté. On peut remplacer les instances de mailx par s-nail dans les scripts car mailx est remplacé par s-nail dans les mises à jour les plus récentes - voir ce lien en anglais https://www.systutorials.com/sending-email-from-mailx-command-in-linux-using-gmails-smtp/. Le script doit maintenant être exécuté de façon automatique à intervalle régulier, nous utiliserons crontab qui a déjà été installé précédemment avec le paquet cronie. On prépare crontab de la façon suivante : `crontab -e`

```
# Edit this file to introduce tasks to be run by cron.
#.
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#.
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#.
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
```

```
#.
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#.
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#.
# For more information see the manual pages of crontab(5) and cron(8)
#.
# m h  dom mon dow   command
01  */12   *   *   *   ~/.config/verif_IP >/dev/null 2>&1
```

Ajoutez la dernière ligne (copiez-collez) et quitter en sauvegardant en tapant `:q` (c'est la façon de quitter de l'éditeur `nano`, acceptez d'écrire avec `o` et le fichier `crontab` sera édité et actif. Cette ligne signifie qu'à chaque première minute de l'heure (01) à chaque 12 heures (*/12), à tous les jours, à tous les mois, à chaque jour de la semaine, on exécute la commande `~/.config/verif_IP`. Le ~, pour rappel, représente votre répertoire /home/usager. /dev/null est pour vous empêcher de recevoir un courriel interne de crontab à chaque exécution du script (soit 2 fois par jour). Ainsi votre adresse IP sera vérifiée à minuit et 1 minute et à midi et une minute à tous les jours. Si vous êtes loin du serveur vous n'aurez jamais plus de 12 heures à attendre pour obtenir votre nouvelle adresse dans votre courriel. Vous pouvez ajuster selon vos besoins à une fréquence plus élevée ou plus basse.

WOL ou Wake-on-LAN

Le nano-ordinateur Raspberry Pi consomme très peu d'énergie et on peut évidemment le laisser fonctionner 24h à l'année. Depuis le pi on peut réveiller (Wake-on-LAN) n'importe quel ordinateur **branché en filaire** au routeur avec lequel communique le pi. Ainsi depuis le pi à distance on peut démarrer un autre appareil, s'y connecter ensuite par ssh, y effectuer les tâches désirées et enfin le fermer à distance. Il faut utiliser la fonction *Wake-on-LAN* du logiciel éponyme `wakeonlan` (voir dans la liste des logiciels pratiques que je vous ai fait installer). Presque tous les ordinateurs possèdent cette possibilité de réveil par le réseau : un *nombre magique* est transmis à la carte de réseau (identifiée par son adresse MAC) pour forcer l'allumage.

- Pour savoir si la carte réseau est compatible WOL il faut d'abord connaître le nom du périphérique de la carte de réseau avec la commande `ip addr` et regarder le nom qui est suivi des mots `link/ether`, ce sera souvent `enp3s0`;
- On lance ensuite la commande suivante : `sudo ethtool enp3s0 | grep Wake-on`. Si la commande répond ce qui suit, votre ordinateur possède la fonction WOL et peut être réveillé à distance :
  `Supports Wake-on: pumbg`
  `Wake-on: g`
  Il faut que la lettre `g` (pour *magic packet activity*) apparaisse sur la ligne du support et aussi à la ligne Wake-on.
- Si une autre lettre que g apparaît à la ligne `Wake-on`, alors allez voir les solutions à https://wiki.archlinux.org/index.php/Wake-on-LAN#Enable_WoL_on_the_network_adapter
- La commande pour lancer à distance un autre appareil se fait depuis le Raspberry Pi mais en utilisant l'adresse matérielle MAC de la carte de réseau à réveiller. Dans la commande `ip addr` du

point 1 il faut noter le sextuplet qui suit les mots `link/ether`, par exemple `1c:60:9c:d7:d5:fe`. Le logiciel `wakeonlan` utilisera l'adresse matérielle de votre appareil à réveiller (et non pas son adresse IP) pour transmettre un numéro magique.
- Depuis le Raspberry Pi il suffit de taper : `wakeonlan 1c:60:9c:d7:d5:fe` pour réveiller cet ordinateur.
- `ssh -Y monordi` pour s'y brancher et, quand on a terminé, `sudo shutdown -h now` pour le fermer. Évidemment avant de tenter de vous brancher par `ssh` laissez le temps à l'appareil de s'ouvrir!

Le script suivant, que j'ai mis dans le répertoire `/usr/local/bin` du Raspberry Pi, permet de réveiller des appareils en tapant simplement `reveille` à la ligne de commande. On tape ensuite la lettre correspondant à l'ordi à réveiller. Le script, nommé `reveille`, a ensuite les droits suivants : `sudo chmod 700 /usr/local/bin/reveille ; sudo chown usager:usager /usr/local/bin/reveille` Voici le script à adapter selon vos besoins :

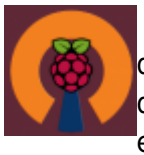
```
#!/bin/bash
# définition des adresses MAC
monordi=1c:60:9c:d7:d5:fe
votrordi=61:24:47:1b:f2:2a
echo "Quel PC faut-il réveiller ?"
echo "m) monordi"
echo "v) votrordi"
echo "q) quitter"
read input1
case $input1 in
  m)
    /usr/bin/wakeonlan $monordi
    ;;
  v)
    /usr/bin/wakeonlan $votrordi
    ;;
  Q|q)
    exit
    ;;
esac
```

!

Deux autres serveurs : PiVPN et Pi-hole

Comme accéder à son réseau grâce à un Raspberry Pi exige vraiment très peu de ressources, on peut facilement ajouter deux autres serveurs au nano-ordinateur. Un premier pour faciliter l'accès à notre réseau local, améliorer notre sécurité et celle des périphériques que l'on connecte à notre réseau et un second pour notre confort de navigation dans l'Internet en supprimant les publicités. **PiVPN** http://www.pivpn.io/ est un serveur *OpenVPN* adapté au Raspberry Pi et **Pi-hole** https://pi-hole.net/ est un trou noir qui filtrera les publicités avant qu'elles atteignent les périphériques connectés à votre réseau, permettant en prime d'économiser de la bande passante!

PiVPN



PiVPN offre une installation simplifiée du protocole *OpenVPN* sur un Raspberry Pi. Un script d'installation depuis l'Internet vous permet de créer votre propre serveur OpenVPN en une dizaine de minutes. Il y a toutefois un hic : pour que votre serveur fonctionne sans collision ou erreur d'adressage, vous devrez d'abord modifier les adresses internes de votre réseau *en évitant les classiques* 192.168.x.y.

Pourquoi? La presque totalité des modem/routeurs utilisent par défaut la gamme 192.168.x.y, ce qui ne pose aucun problème pour votre réseau domestique. Mais si vous installez un serveur OpenVPN et que vous y accédez depuis une connexion wifi (publique) hors de votre réseau, il y a fort à parier que le wifi public sert déjà des clients *dans la même gamme* 192.168.x.y. Alors le serveur OpenVPN ne peut pas distinguer les clients des deux réseaux aux adresses similaires, il y a des collisions.

La solution est de modifier ses adresses internes en préférant une gamme moins susceptible d'être utilisée par les autres modem/routeurs. L'IANA (Internet Assigned Numbers Authority) autorise les gammes suivantes comme adresses privées :
192.168.0.0 à 192.168.255.255 pour plus de 65 000 possibilités d'adresses;
172.16.0.0 à 172.31.255.255 pour plus de 1 million d'adresses;
10.0.0.0 à 10.255.255.255 pour plus de 16 millions d'adresses.
Choisissez une gamme du genre `10.33.66.100` et vous serez assuré d'éviter toute collision!
Évidemment, selon votre routeur, *vous devrez faire les changements vous-mêmes*. Faites toujours des saisies d'écran des différents paramètres de votre réseau ou, plus simplement, sauvegardez sa configuration complète – habituellement dans la configuration avancée de votre routeur. Servez-vous de la documentation de votre routeur, vous la trouverez facilement dans l'Internet!

Prenez une adresse IP semblable à `10.33.66.1` et dans les paramètres du serveur DHCP débutez la pile par `10.33.66.100`. Réservez des adresses pour vos périphériques et, si vous le désirez, modifiez les fichiers `/etc/hosts` de vos ordinateurs. Faites aussi un transfert de port pour OpenSSH pour ajouter une couche de sécurité.

L'installation de PiVPN se fait depuis l'Internet après s'être connecté en ssh sur le Raspberry Pi. Je vous suggère, avant de commencer l'installation, la lecture (en anglais) du site suivant qui vous montre avec des saisies d'écran les différentes étapes :
http://kamilslab.com/2017/01/22/how-to-turn-your-raspberry-pi-into-a-home-vpn-server-using-pivpn/.

Prêt? En ssh sur votre Raspberry Pi, comme vous êtes déjà en ligne de commande, tapez (faites un copier-coller) :
```
curl -L https://install.pivpn.io | bash
```
Suivez les directives (en anglais), les valeurs par défaut sont habituellement correctes. Pour ma part j'ai choisi OpenDNS comme serveur DNS et un port différent de 1194 pour le VPN.

Quand l'installation sera complétée, vous créerez des fichiers `ovpn` pour chacun de vos clients avec la commande `pivpn add` et vous copierez ces fichiers sur vos périphériques (par `ftp` ou à la rigueur par courriel). Un fichier par client/usager différent.

Selon le bureau/SE (système d'exploitation) utilisé, vous n'aurez qu'à importer le fichier ovpn pour établir la connexion « permanente » sécurisée avec votre Raspberry Pi et votre ordinateur portable. Le principe est le même avec un cellulaire, OpenVPN a une interface cliente facile à utiliser.

Vous ne pourrez pas utiliser le VPN se vous êtes déjà dans votre propre réseau interne : si votre cellulaire est branché en wifi chez vous, vous ne pourrez pas utiliser le VPN parce que vous êtes déjà dans votre propre réseau. Pour tester, *désactivez le wifi de votre cellulaire* pour qu'il se connecte à l'Internet hors de votre réseau.

Que faire si l'adresse IP du serveur OpenVPN change?

Le script `~/.config/verif_IP` qui est exécuté 2 fois par jour (à 0h01 et 12h01) vérifie notre adresse IP, celle où réside notre serveur OpenVPN. Lorsque l'adresse change, nous recevons la nouvelle adresse par courriel. Lorsque cela arrive, il faut changer les fichiers `.ovpn` de nos périphériques qui utilisent notre VPN (cellulaires, portable, etc).

Pour ce faire :

1. Éditer `/etc/openvpn/easy-rsa/pki/Default.txt` pour que la ligne reflétant l'adresse du serveur soit la bonne
2. Exécuter la commande `pivpn -l` pour lister les clients OpenVPN enregistrés
3. Regénérer les fichiers `.ovpn` avec la commande `pivpn add`
4. Au besoin, révoquez l'ancien client avec `sudo pivpn -r ancienclient`
5. Exportez chacun des nouveaux fichiers `.ovpn` sur les périphériques respectifs, ajoutez les nouveaux clients et supprimez les anciens.

Chez Vidéotron, ma dernière adresse IP a tenu 14 mois avant d'être changée!

Pi-hole

Pi-hole est un trou noir pour les publicités de l'Internet; il filtre en amont avec une impressionnante efficacité. Vous pourrez même ensuite désactiver le bloqueur de publicité de votre fureteur (*UblockOrigin* par exemple). Certes quelques publicités peuvent peut-être encore s'infiltrer, mais vous aurez l'avantage d'apparaître sans bloqueur pour les serveurs externes 😉 !

Le serveur est très léger et, si vous l'utilisez pour y faire transiter tout l'Internet de votre réseau domestique, tous les clients (appareils connectés en filaire ou wifi) bénéficieront de Pi-hole. Et c'est aussi valable hors de votre réseau si vous utilisez votre VPN pour utiliser l'Internet!

L'installation est similaire à celle de PiVPN : connectez-vous sur le Raspberry Pi en ssh, tapez (copier-coller) ensuite la commande suivante :
```
curl -sSL https://install.pi-hole.net | bash
```

Votre Raspberry Pi redémarrera à la fin du processus et le serveur Pi-hole sera opérationnel.

Quand Pi-hole est installé on modifie la configuration de notre routeur pour que notre Raspberry Pi (avec Pi-hole) soit **notre unique serveur DNS**. C'est la seule façon de s'assurer que tout notre trafic Internet sera filtré par Pi-hole. La FAQ (foire aux questions) du site Internet de Pi-hole présente (en anglais) 3 méthodes pour le faire, la première et la plus simple est généralement suffisante :

https://discourse.pi-hole.net/t/how-do-i-configure-my-devices-to-use-pi-hole-as-their-dns-server/245.

Allez dans la configuration avancée de votre routeur, dans ses options DHCP, et renseignez l'adresse de votre Raspberry Pi comme seul serveur DNS. Supprimez (videz) les adresses obtenues du FAI (fournisseur d'accès Internet). Appliquez les changements et au besoin redémarrez votre routeur.

Vous pourrez accéder à l'administration et à la configuration du serveur Pi-hole en tapant son adresse dans votre fureteur. Dans ses paramètres (*Settings* - son interface est en anglais), sous *System* sélectionnez *Disable query logging* pour minimiser les écritures sur la carte microSD du Raspberry Pi.

Symbiose PiVPN + Pi-hole

Nos serveurs PiVPN et Pi-hole fonctionnent mais le fin du fin est qu'on peut les faire fonctionner ensemble! L'avantage est nul pour votre réseau local, lorsque vous êtes chez vous, puisque de toute façon vous ne pouvez pas vous connecter en local avec PiVPN. Mais si vous êtes hors du rayon d'action de votre réseau local, vous pourrez utiliser PiVPN pour vous connecter chez vous et profiter simultanément de Pi-hole.

- Ainsi d'un wifi public vous pourrez vous brancher avec la sécurité de votre réseau interne et profiter du filtrage des publicités de Pi-hole;
- Depuis un réseau wifi externe, Pi-VPN vous garantira l'adresse IP de votre modem/routeur de la maison ainsi que sa sécurité même hors du pays.

Il reste 3 courtes étapes (ce qui suit est inspiré et traduit librement du site Internet https://marcstan.net/blog/2017/06/25/PiVPN-and-Pi-hole/) pour relier les serveurs PiVPN et Pi-hole.

Connectez-vous en ssh sur votre Raspberry Pi :

1. Éditez, comme root, le fichier de configuration du serveur OpenVPN
   `sudo nano /etc/openvpn/server.conf` et mettez en commentaire (en ajoutant un # au début de la ligne) toutes les lignes où vous trouverez `dhcp-option` puis ajoutez cette ligne
   `push "dhcp-option DNS 10.8.0.1"`
   `Ctrl + x` pour quitter et enregistrer, `o` pour accepter, `[Entrée]` pour valider (vous venez d'utiliser l'éditeur texte nano comme super-utilisateur)

2. Éditez, comme root, le fichier de configuration de Pi-hole
   `sudo nano /etc/pihole/setupVars.conf`
   Ajoutez "`PIHOLE_INTERFACE=tun0`" sous celle qui contient "`eth0`".
   Vous devriez maintenant retrouver ceci :

   ```
   PIHOLE_INTERFACE=eth0
   PIHOLE_INTERFACE=tun0
   ```

   `Ctrl + x` pour quitter et enregistrer, `o` pour accepter, `[Entrée]` pour valider.
   Évidemment `eth0` vaut pour une connexion ethernet de votre Raspberry Pi, ajustez pour `wlan0` si jamais votre Pi était plutôt configuré pour atteindre l'Internet uniquement en wifi.

3. Nous allons enfin créer un dernier fichier (car il ne devrait pas être présent) :
   `sudo nano /etc/dnsmasq.d/02-ovpn.conf`
   qui ne contiendra que la ligne suivante :
   “`interface=tun0`”
   Puis `Ctrl + x` pour quitter et enregistrer, `o` pour accepter, `[Entrée]` pour valider.

Voilà, on redémarre le Raspberry Pi (`sudo reboot`) et on profite de l’Internet 😉 !

Attention à la mise-à-jour de Pi-hole

La console d’administration en mode graphique de Pi-hole (http://adresse-du-pi/admin) peut indiquer qu’une mise-à-jour est disponible. C’est tout simple : `sudo pihole -up` sur le Raspberry Pi et tout se déroule sans même redémarrer, enfin… presque! Comme nous avons fait le lien (symbiose) OpenVPN-Pi-hole précédemment, deux fichiers de configuration importants sont malheureusement **écrasés** avec la mise à jour et vos périphériques **vont perdre l’Internet…**

Il faut corriger les deux fichiers suivants (après la mise-à-jour de Pi-hole) :

1. le fichier `/etc/dnsmasq.d/01-pihole.conf` doit contenir à la dernière ligne `interface=eth0`
2. le fichier `/etc/pihole/setupVars.conf` doit contenir les **deux** lignes suivantes :

```
PIHOLE_INTERFACE=eth0
PIHOLE_INTERFACE=tun0
```

Après quoi il faudra redémarrer le Raspberry Pi.