

Master 2^{nde} année

TMC

Raspberry Pi

Raspberry Pi: démarrage par réseau et montage NFS

Préparation du démarrage bootp, PXE

On va créer le répertoire RASPI, dédié au «filesystem » du Raspberry PI:

□ le répertoire occupera autour de 3Go en utilisation :

\$ mkdir RASPI \$ cd RASPI	—

- un répertoire client contenant l'intégralité du système Raspbian du Raspberry (répertoires /etc, /home, /bin, etc.) qui sera accessible par le protocole NFS;
- □ un répertoire boot contenant le noyau et les fichiers de « *bas-niveau* » pour le raspberry Pi lui-même, qui sera accessible par le protocole bootp;

Vous téléchargerez la distribution «Raspbian lite » sur le site officiel du Raspberry PI et le mettrez dans votre répertoire RASPI.

xterm	
<pre>\$ wget https://downloads.raspberrypi.com/raspios_lite_arm64/images/ras</pre>	
pios_lite_arm64-2023-12-11/2023-12-11-raspios-bookworm-arm64-lite.img.xz	
\$ xz -d 2023-12-11-raspios-bookworm-arm64-lite.img.xz	

Nous récupérerons le contenu des deux partitions de cette distribution pour remplir nos deux répertoires :

▷ le filesystem raspbian dans le répertoire client depuis la partion nº2:

```
pef@cube:~ $ sudo losetup -fP 2023-12-11-raspios-bookworm-arm64-lite.img
pef@cube:~ $ losetup -a | grep rasp
/dev/loop10: []: (/home/pef/2023-12-11-raspios-bookworm-arm64-lite.img)
pef@cube:~ $ sudo mount /dev/loop10p2 /mnt
pef@cube:~ $ mkdir RASPI
pef@cube:~ $ cd RASPI
pef@cube:/~RASPI $ mkdir client
pef@cube:/~RASPI $ sudo rsync -xa --progress /mnt/ client/
pef@cube:/~RASPI $ sudo umount /mnt
```

 \triangleright les fichiers de « boot » depuis la partition nº1 :

Nous installerons le serveur NFS :

pef@cube:~/RASPI \$ sudo apt install nfs-kernel-server

Configuration du partage NFS dans le fichier /etc/exports:

```
pef@cube:/etc $ cat exports
# /etc/exports: the access control list for filesystems which may be exported
# to NFS clients. See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes hostname1(rw,sync,no_subtree_check)
hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4 gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)
#
/home/pef/RASPI/client *(rw,sync,no_subtree_check,no_root_squash)
/home/pef/RASPI/boot *(rw,sync,no_subtree_check,no_root_squash)
```

On active le service NFS et RPCBind :

п-

- xterm ·

xterm .

```
pef@cube:~/RASPI $ sudo systemctl enable nfs-kernel-server
pef@cube:~/RASPI $ sudo systemctl enable rpcbind
```

Si vous modifiez la configuration d'un export, il faut redémarrer le service NFS :

pef@cube:~/RASPI \$ sudo systemctl restart nfs-kernel-server

Pour voir les points de montage offert par un serveur NFS :

```
xterm
$ showmount -e 127.0.0.1
Export list for 127.0.0.1:
/home/pef/RASPI/boot *
/home/pef/RASPI/client *
```

Mise en service du serveur TFTP, DNS, DHCP

On va utiliser la commande dnsmasq dans le script script_boot_rpi:

```
#!/bin/bash
# interface du PC connexion Raspberry
IF=enpls0
# pour un menu de sélection de l'interface avec fzf
# IF=$(command -v fzf > /dev/null 2>&1 && ip 1 | awk '/^[0-9]+/ { print substr($2, 1, length($2)-1)}' | fzf)
PREFIX=10.20.30
sudo nmcli device set $IF managed no
sudo sysctl -w net.ipv4.ip_forward=1
sudo ip link set dev $IF down
sudo ip link set dev $IF wn
sudo ip link set dev $IF $PREFIX.1/24
sudo ip tables -t nat -F
sudo iptables -t nat -A POSTROUTING -s $PREFIX.150,255.255.0,12h -O 3,$PREFIX.1
-O 6,8.8.8.8 --pxe-service=0, "Raspberry Pi Boot" --enable-tftp --tftp-root=$HOME/RASPI/boot
```

Montage de NFS sur le Raspberry Pi

On modifie le point de montage du Raspberry Pi pour son filesystem, en éditant le fichier

~/RASPI/boot/cmdline.txt xterm
console=serial0,115200 console=tty1 root=/dev/nfs
nfsroot=10.20.30.1:/home/pef/RASPI/client,tcp,vers=3 rw ip=dhcp rootwait Attention: c'est tout sur une seule ligne. On modifie le fichier etc/fstab du Raspberry Pi en:
/RASPI/client/etc/fstab:

proc	/proc	proc	de	efaults		0	0	1
10.20.30.1	:/home/pef/RASPI/boot	/boot	nfs	defaults,	vers=3	0	0	

Il ne doit contenir que ces deux lignes.

Resp. UE : P-F. Bonnefoi, http://p-fb.net/, «TMC-Raspberry Pi» version du 19 décembre 2023, rédigé avec ConTEXt – Don't Panic !

```
Création du mot de passe pour l'utilisateur pi
```

```
$ sudo apt install whois
$ mkpasswd -m sha-512 -S UMsalt12 raspberry
$6$UMsalt12$qw0IZGjzlp3hDux.unsQoA6C16B7byh/T49k6VW8tHNnzBsIj5iW
NYT9Ra.ObN9XKYQ51ygiubNf2GKU1dwoA.
```

On modifie le fichier etc/shadow du système du raspberry pi:

```
pef@cube:~/RASPI/client $ cat shadow | grep pi
pi:$6$UMsalt12$qw0IZGjzlp3hDux.unsQoA6C16B7byh/T49k6VW8tHNnzBsIj5iW
NYT9Ra.ObN9XKYQ51ygiubNf2GKU1dwoA.:19702:0:99999:7:::
```

Activation du service SSH sur le Raspberry PI

On passe par le point de montage NFS, c-à-d le répertoire local correspondant au filesystem NFS :

```
xterm
pef@cube:~ $ cd RASPI/client
pef@cube:~/RASPI/client $ cd etc/systemd/system
pef@cube:~/RASPI/client/systemd/system $ sudo ln -s
/lib/systemd/system/sshswitch.service .
pef@cube:~/RASPI/client $ sudo vi lib/systemd/system/sshswitch.service
```

<u>et modifier ce fichier en :</u>

xterm

xterm

```
xterm
[Unit]
Description=Turn on SSH if /boot/ssh is present
After=regenerate_ssh_host_keys.service
[Service]
Type=oneshot
ExecStart=/bin/sh -c "systemctl enable --now ssh"
[Install]
```

WantedBy=multi-user.target

xterm

ln.

On peut maintenant procéder à :

- ▷ l'allumage du Raspberry Pi.
- lancement du script de prise en charge du Raspberry Pi : \triangleright

pef@cube:~/RASPI \$ sudo ./script_boot_rpi

Dans la sortie du script de boot on a :

```
xterm
D-
    _
 ./script_boot_rpi
net.ipv4.ip_forward = 1
RTNETLINK answers: File exists
dnsmasq: started, version 2.89 cachesize 150
dnsmasq: compile time options: IPv6 GNU-getopt DBus no-UBus i18n IDN2 DHCP DHCPv6 no-Lua TFTP conntrack ipset nftset auth cryptohash DNSSEC loop-detect inotify dumpfile dnsmasq-dhcp: DHCP, IP range 10.20.30.100 -- 10.20.30.150, lease time 12h
dnsmasq-dhcp: DHCP, sockets bound exclusively to interface enpls0
dnsmasq-tftp: TFTP root is /home/pef/RASPI/boot
dnsmasq: reading /etc/resolv.conf
dnsmasq: using nameserver 127.0.0.53#53
dnsmasq: read /etc/hosts - 7 addresses
dnsmasq-dhcp: DHCPDISCOVER(enx000ec6885a80) b8:27:eb:30:2d:0a
dnsmasq-dhcp: DHCPOFFER(enx000ec6885a80) 10.20.30.144 b8:27:eb:30:2d:0a
dnsmasq-tftp: sent /home/pef/RASPI/boot/bootcode.bin to 10.20.30.144
dnsmasq-tftp: sent /home/pef/RASPI/boot/start.elf to 10.20.30.144
dnsmasg-tftp: sent /home/pef/RASPI/boot/cmdline.txt to 10.20.30.144
dnsmasq-tftp: sent /home/pef/RASPI/boot/cmdline.txt to 10.20.30.144
dnsmasg-tftp: file /home/pef/RASPI/boot/recovery8.img not found
dnsmasq-tftp: failed sending /home/pef/RASPI/boot/kernel8.img to 10.20.30.144
dnsmasq-tftp: file /home/pef/RASPI/boot/kernel8-32.img not found
dnsmasq-tftp: sent /home/pef/RASPI/boot/kernel7.img to 10.20.30.144
dnsmasq-dhcp: DHCPDISCOVER(enx000ec6885a80) b8:27:eb:30:2d:0a
dnsmasg-dhcp: DHCPOFFER(enx000ec6885a80) 10.20.30.144 b8:27:eb:30:2d:0a
dnsmasq-dhcp: DHCPREQUEST(enx000ec6885a80) 10.20.30.144 b8:27:eb:30:2d:0a
dnsmasq-dhcp: DHCPACK(enx000ec6885a80) 10.20.30.144 b8:27:eb:30:2d:0a
dnsmasq-dhcp: DHCPDISCOVER(enx000ec6885a80) b8:27:eb:30:2d:0a
dnsmasq-dhcp: DHCPOFFER(enx000ec6885a80) 10.20.30.144 b8:27:eb:30:2d:0a
dnsmasq-dhcp: DHCPREQUEST(enx000ec6885a80) 10.20.30.144 b8:27:eb:30:2d:0a
dnsmasq-dhcp: DHCPACK(enx000ec6885a80) 10.20.30.144 b8:27:eb:30:2d:0a raspberrypi
```

Resp. UE : P-F. Bonnefoi, http://p-fb.net/, «TMC–Raspberry Pi » version du 19 décembre 2023, rédigé avec ConT_FXt – Don't Panic !

Lorsque le Raspberry Pi envoi son "raspberrypi" dans le DHCPACK c'est qu'il est passé sous Raspian. À ce moment là, vous devez pouvoir vous connecter par ssh :

	xterm -	-
\$ ssh	pi@10.20.30.144	l

Astuces

IL est conseillé de relancer votre script offrant le DNS/tftp/DHCP à votre Raspberry Pi dès que vous l'allumez ou après l'avoir éteint/rallumé : en effet, lorsque le Raspberry Pi est éteint, le câble ethernet reliant votre PC au Raspberry Pi n'est plus alimenté par le Rapsberry Pi et votre PC désactive son interface automatiquement ce qui bloque votre script.

Si votre Raspberry Pi est connecté directement à votre ordinateur sans utiliser de switch, il peut être nécessaire de désactiver/réactiver votre interface réseau :

pef@cube:~/RASPI \$ sudo ip 1 set enx00606e20053f down;sudo ip 1 set enx00606e20053f up

En effet, quand vous débranchez votre Raspberry Pi, son interface réseau n'est plus alimenté, ce qui peut amener votre ordinateur à faire « tomber » son interface réseau (sans que l'état indiqué par la commande ip link le montre).

Configuration du Raspberry Pi

Pour les mises à jour :

🔲 — xterm

🔲 — xterm pi@raspberrypi:~ \$ sudo apt update pi@raspberrypi:~ \$ sudo apt upgrade pi@raspberrypi:~ \$ sudo reboot

Il est conseillé de relancer le script de prise en charge du Raspberry Pi (dnsmasq).

Ensuite, on reprend la configuration du Raspberry Pi.

La configuration du pays pour le WiFi:

```
🔲 — xterm -
pi@raspberrypi:~ $ rfkill unblock all
pi@raspberrypi:~ $ wpa_cli -i wlan0 set country FR
pi@raspberrypi:~ $ wpa_cli -i wlan0 save_config
```

Pour la configuration du point d'accès :

```
___ xterm
pi@raspberrypi:~ $ sudo apt install hostapd dnsmasq iptables
pi@raspberrypi:~ $ sudo systemctl disable dnsmasq
pi@raspberrypi:~ $ sudo systemctl stop dnsmasq
```

Pour la mise en place d'un point d'accès, le contenu du fichier script_ap:

#!/bin/bash
INTERFACEWAN=eth0
INTERFACE=wlan0
SSID=10T le SSID de votre point d'accès à personnaliser
PSR=12344321
PREF1X=10.33.33
CFGHOSTAPD=/tmp/hostapd_config
cat < <end> \$CFGHOSTAPD</end>
interface=\$INTERFACE
hw_mode=g
macaddr_acl=0
auth_algs=3
channel=6
ssid=\$SSID
wpa=2
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
wpa passphrase=\$PSK
sudo ip 1 set dev \$INTERFACE down
sudo in l set dev SINTERFACE un
sudo ip a flush dev SINTERFACE
sudo ip a add \$PREFIX.254/24 dev \$INTERFACE
sudo killall hostand
sudo hostand SCEGHOSTAPD &
sude suscit - what invit in forward-1
Sudo Systel w het. pv3.1p_101walu-1
Sudo iptables - that - A POSIKOUTING -S \$PREFIX.0/24 -0 \$INTERFACEWAN -] MASQUERADE
sudo dnsmasq -d -z -a \$PREFIX.254 -F \$PREFIX.100,\$PREFIX.150,255.255.255.0 -O 6,\$PRE FIX.254 -A /serveur.iot.com/\$PREFIX.254 -1 /tmp/leases

pi@raspberrypi:~ \$ chmod +x script_ap
pi@raspberrypi:~ \$ sudo ./script_ap

Vous devriez pouvoir vous connecter en WiFi sur le Raspberry Pi depuis l'ESP8266 ou une autre machine.

Vous pourrez faire une requête DNS vers serveur.iot.com qui devrait renvoyer 10.33.33.254, c-à-d le Raspberry Pi lui-même où devra tourner le serveur MQTT.

ATTENTION

Vous devez changer le SSID dans le script sinon vous allez vous connecter sur l'AP du voisin...