

Beste de savoir

Installations automatisées de machines virtuelles avec libvirt

15 janvier 2019

Table des matières

1.	Introduction	1
2.	Petit aperçu de libvirt	1
3.	Petit aperçu de mon infrastructure	3
4.	virt-install, mon amour	4
5.	Le script complet	7
5.1.	Petit bonus : installation automatique d'une clé SSH	9
6.	Conclusion	9
	Contenu masqué	9

% INSTALLATIONS AUTOMATISÉES DE MACHINES VIRTUELLES AVEC LIBVIRT %
Sandhose % 27 janvier 2018

1. Introduction

Je m'occupe depuis quelque temps de l'infrastructure de l'Amicale des Informaticiens de l'Université de Strasbourg (AIUS, pour faire court), qui est une association étudiante de l'UFR de Mathématique et Informatique à l'Université de Strasbourg.

On a donc un local, avec un canapé, des snacks et de la bonne humeur ; mais aussi une baie¹ avec 2 serveurs et un switch. Sur ces serveurs on y héberge des boîtes mail, des listes de diffusion, divers sites internet et autres machines virtuelles à disposition des étudiants.

Je me suis dit qu'il serait intéressant de détailler quelques travaux et expérimentations que j'ai pu y faire. Je ne sais pas si je ferais toute une série de billets, ou si celui-ci sera le seul que j'arriverais à écrire, mais je me lance.

2. Petit aperçu de libvirt

J'ai donc deux machines sous Debian 9, `tic` et `tac`. Aucun service ne tourne directement sur ces machines, mis à part `libvirt`, pour faire tourner un certain nombre de machines virtuelles.

`libvirt` est une couche d'abstraction autour de divers systèmes de virtualisation tels que `qemu` (`kvm`), Virtual Box, Xen, LXC (conteneurs), HyperV... Le but est de proposer une API unifiée pour tous ces outils. Aussi, personne ne veut écrire de commande `qemu` à la main.

👁️ Contenu masqué n°1

1. Je la prendrai en photo à l'occas.

2. Petit aperçu de libvirt

`libvirt` se présente comme un démon, `libvirtd`, qui tourne sur l'hyperviseur offrant une API utilisée ensuite par une multitude d'outils.

`virt-manager` par exemple propose une interface graphique permettant de se connecter à plusieurs démons `libvirtd`.

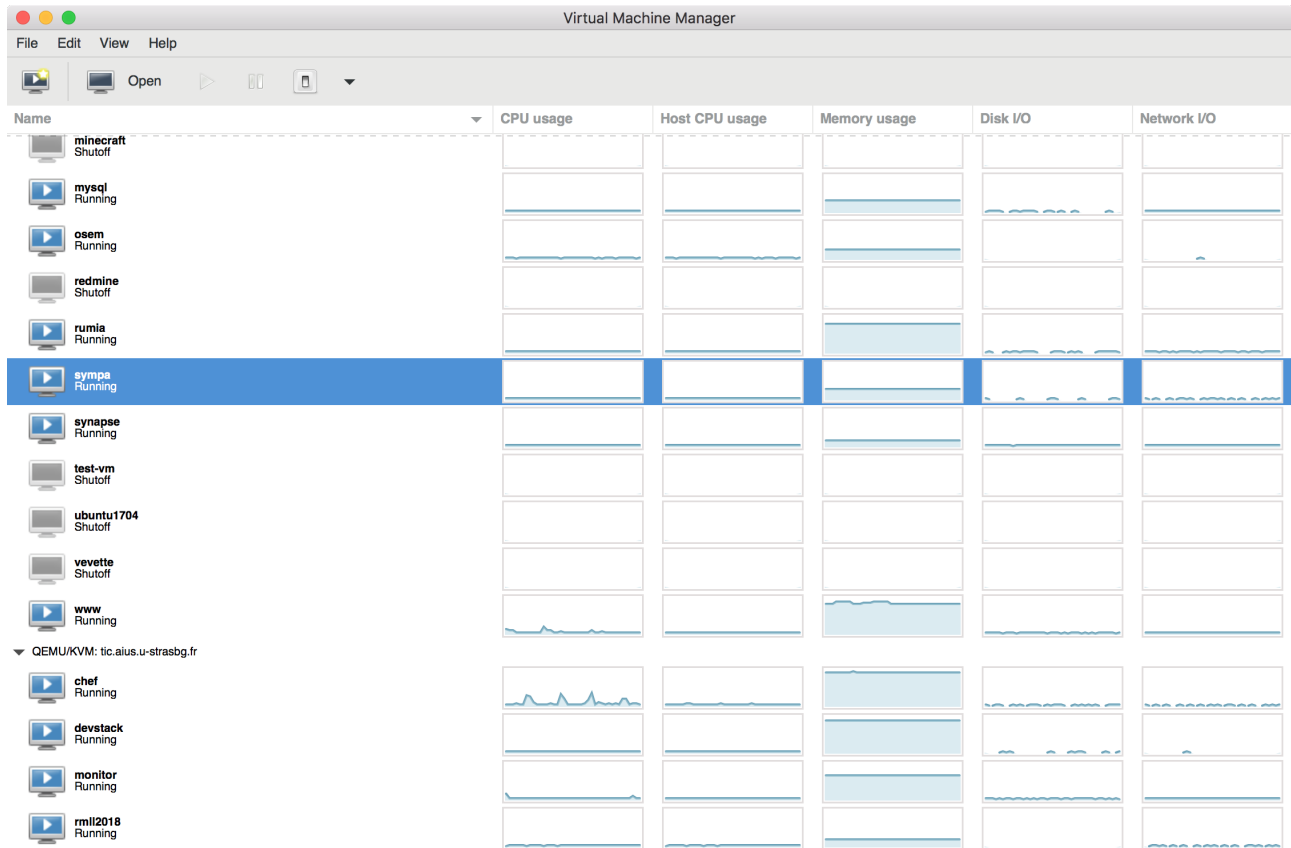


FIGURE 2. – Extrait de la liste des VM tournant sur 'tic' et 'tac'

3. Petit aperçu de mon infrastructure

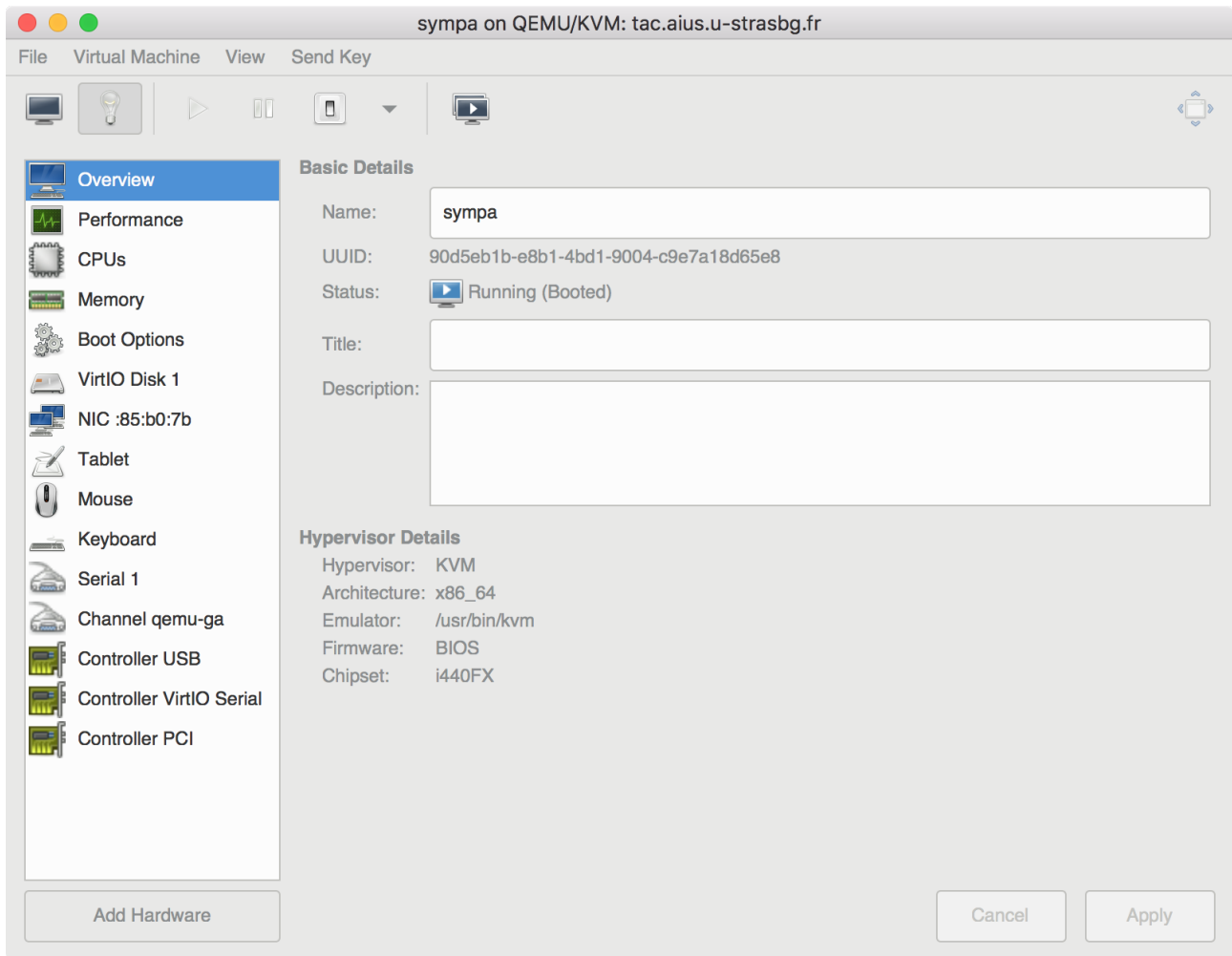


FIGURE 2. – Interface de configuration d'une machine virtuelle

`virsh` est un outil en ligne de commande pour gérer les machines tournant via `libvirtd`. Tout ce qui est possible (ou presque) avec `virsh` l'est aussi avec `virt-manager` et vice-versa. Ce sont les deux outils de référence développés par l'équipe de `libvirt`.

On va parler plus loin de `virt-install`, qui est un outil permettant de scripter facilement l'installation de machines virtuelles en ligne de commande.

3. Petit aperçu de mon infrastructure

Je vais m'étendre un chouilla sur l'aspect surtout réseau de mon infrastructure. C'est assez spécifique à mon utilisation, donc si ça vous intéresse pas, vous pouvez sauter complètement cette partie.

À l'inverse, si mes aventures réseau vous intéresse, je peux complètement détailler ça dans un autre billet.

La `DI` de ma fac nous laisse à disposition tout le sous-réseau 130.79.49.192/27 en v4 (donc 32 IPs)² + un /64 en v6.

4. *virt-install*, mon amour

J'aurais pû assigner ces IPs directement au machines virtuelles via un bridge, mais je préfère avoir un réseau privé avec mes machines virtuelles (10.0.0.0/16, pour faire dans l'originalité), et une machine qui sert de NAT + DHCP + serveur DNS interne.

Cette machine s'appelle `gateway`, et tourne sous OpenBSD (parce que OpenBSD (et `pf`) ça roxxe). C'est la seule machine qui a accès aux deux réseaux, le réseau publique et privé.

Elle a un certain nombre d'IP, et redirige différents ports vers différentes VMs

☉ Contenu masqué n°2

Cette machine fait aussi tourner un serveur DHCP pour les machines virtuelles, ainsi qu'un serveur DNS local (`unbound`), pour la zone `int.aius.u-strasbg.fr`.

Un petit extrait de la configuration `unbound` pour vous donner une idée de son utilité :

```
1 local-zone: "int.aius.u-strasbg.fr." static
2 local-data: "gateway.int.aius.u-strasbg.fr. IN A 10.0.0.1"
3 local-data: "switch.int.aius.u-strasbg.fr. IN A 10.0.0.2"
4 local-data: "tic.int.aius.u-strasbg.fr. IN A 10.0.0.3"
5 local-data: "tac.int.aius.u-strasbg.fr. IN A 10.0.0.4"
6 local-data: "ldap.int.aius.u-strasbg.fr. IN A 10.0.1.1"
7 local-data: "www.int.aius.u-strasbg.fr. IN A 10.0.1.2"
8 local-data: "mysql.int.aius.u-strasbg.fr. IN A 10.0.1.3"
9 local-data: "mail.int.aius.u-strasbg.fr. IN A 10.0.1.4"
```

Le serveur DHCP annonce comme domaine de recherche `int.aius.u-strasbg.fr`. De cette manière, toutes les machines sont accessibles « entre elles » avec leur nom court. Je peux donc faire à tout moment `ssh www` pour me connecter sur la VM `www` depuis n'importe quelle VM (modulo la clé SSH/le mot de passe).

De même, aucuns de mes fichiers de configuration n'utilise d'IP en dur. La machine `www` va contacter la machine `mysql` simplement par son hostname réduit.

Bref, ayez un serveur DNS local, et des noms courts et reconnaissables pour vos machines : ça rend le déplacement et les réorganisations (genre, changement d'IP d'une machine) beaucoup plus simples.

4. *virt-install*, mon amour

`virt-install` permet donc de créer une machine virtuelle en ligne de commande. Donnez-lui quelques infos et une image ISO, il va allouer les ressources nécessaires et démarrer la machine virtuelle sur l'image que vous lui avez donné.

2. Ouais, ils ont beaucoup d'IPv4 à l'Unistra.

4. *virt-install*, mon amour

```
1 virt-install \  
2   --name debian9 \  
3   --ram 1024 \  
4   --disk pool=default,size=10,format=qcow2,bus=virtio \  
5   --vcpus 1 \  
6   --network bridge=virbr0 \  
7   --location  
      https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/debian-9.3.0-  
      amd64-netinst.iso
```

Cela va donc créer une machine virtuelle `debian9` en utilisant l'image d'installation réseau de Debian (on peut lui donner aussi un chemin vers une image locale si besoin), 1 Go de RAM, 10 Go d'espace disque et 1 vCPU.

Sauf que par défaut, il va donner un écran à la machine, et il va donc falloir y accéder via `virt-viewer` pour pouvoir faire quoi que ce soit. Sauf que moi j'aime bien mon terminal, et j'aimerais pouvoir y rester dedans encore un peu.

4.0.1. La console série

On a dû me dire en cours³ à un moment qu'à l'époque, on utilisait des terminaux avec des liaisons séries. Des trucs donc, avec que du texte. Chouette.

Linux prend bien sûr encore en charge cette console série, et l'installateur de Debian fonctionne aussi très bien uniquement avec une console série.

```
1 virt-install \  
2   --name debian9 \  
3   --ram 1024 \  
4   --disk pool=default,size=10,format=qcow2,bus=virtio \  
5   --vcpus 1 \  
6   --network bridge=virbr0 \  
7  
      --location="http://ftp.fr.debian.org/debian/dists/stable/main/installer-a  
      \  
8   --extra-args="auto console=ttyS0,115200n8 serial" \  
9   --graphics none \  
10  --console pty,target_type=serial
```

Et voilà, `virt-install` va afficher tout seul la console série, avec l'installateur Debian :

4. virt-install, mon amour

```
[ (1*installer) 2 shell 3 shell 4- log ][ Jan 27 19:18 ]
| [ !! ] Select a language |
Choose the language to be used for the installation process. The
selected language will also be the default language for the installed
system.
Language:
          C
        English
<Go Back>
<Tab> moves; <Space> selects; <Enter> activates buttons
```

Vous noterez aussi que l'URL a changé : on ne démarre plus sur une image disque (ISO) mais directement [sur un kernel](#) (que libvirt va trouver je ne sais comment). C'est important pour injecter des arguments au démarrage (le `--extra-args`).

4.0.2. Le preseed

Ça c'est la partie fun.

Comme beaucoup de distributions, Debian propose une solution pour automatiser les réponses de l'installateur. On retrouve par exemple kickstart chez Ubuntu, Fedora ou encore CentOS. Sur Debian, il s'agit simplement d'un fichier de réponses pour `debian-installer`.

```
1 $ cat preseed.cfg
2 d-i debian-installer/locale select en_US
3 d-i console-setup/ask_detect boolean false
4 d-i keyboard-configuration/xkb-keymap select fr(latin9)
5
6 $ virt-install \
7   # ...
8     --location="http://ftp.fr.debian.org/debian/dists/stable/main/installer-a
9     \
10    --initrd-inject=preseed.cfg
```

Avec ces trois options, l'installateur va autoconfigurer la langue et la disposition du clavier. Toutes les options de l'installateur sont auto-configurables.

© Contenu masqué n°3

5. Le script complet

Plus qu'à enrober ça dans un petit script shell.

Dans mon infrastructure, il faut rajouter la machine au niveau du serveur DHCP pour qu'elle ait la bonne IP assignée. Pour ça, il me faut l'adresse MAC de l'interface réseau de la machine virtuelle pour pouvoir l'ajouter à la configuration de mon serveur DHCP.

```
1 $ virsh domiflist --domain debian9
2 Interface Type Source Model MAC
3 -----
4 vnet2 bridge virbr0 virtio 52:54:00:f5:6f:a1
5
6 $ virsh domiflist --domain debian9 | awk '/bridge/ { print $5 }'
7 52:54:00:f5:6f:a1
```

Pour l'instant, je me contente d'afficher les instructions à l'écran. À l'avenir j'aimerais que ces changements soient fait automatiquement.

Le script complet :

```
1 #!/usr/bin/zsh
2
3 DOMAIN=$1
4
5 if [ -z "$DOMAIN" ]; then
6     echo "usage: $0 hostname" >&2
7     exit 1
8 fi
9
10 ## On génère le fichier de preseed en remplaçant `HOSTNAME` par le
11    nom de la machine.
12
13 m4 -DHOSTNAME=$DOMAIN < preseed.cfg.in > preseed.cfg
14
15 OPTIONS=(
16     --name=$DOMAIN
17     --vcpus=4 --ram=2048
18
19     --location="http://ftp.fr.debian.org/debian/dists/stable/main/insta
20     --initrd-inject=preseed.cfg
```

3. Je suis en 3^e année de license d'informatique.

5. Le script complet

```
18     --extra-args="auto console=ttyS0,115200n8 serial"
19     --graphics none
20     --console pty,target_type=serial
21     --network bridge=br1
22     --os-type=linux --os-variant=debian9
23     --disk=pool=default,size=50,format=qcow2,bus=virtio
24 )
25
26 echo "|> Running virt-install"
27 virt-install $OPTIONS --noreboot
28
29 reset
30
31 MAC=`virsh domiflist --domain $DOMAIN | awk
32     '/bridge/ { print $5 }'`
33
34 cat - <<EOF
35 |> Install complete. Please configure dhcpd and unbound on gateway
36 - Choose an IP (10.0.1.X)
37 - in /var/unbound/etc/unbound/conf:
38
39     local-data: "$DOMAIN.int.aius.u-strasbg.fr. IN A 10.0.1.X"
40
41 - in /etc/dhcpd.conf:
42
43     host $DOMAIN {
44         hardware ethernet $MAC;
45         fixed-address 10.0.1.X;
46         option host-name "$DOMAIN";
47     }
48
49 - run:
50
51     rcctl restart dhcpd
52     rcctl reload unbound
53
54 EOF
55
56 read -p "Press enter when done."
57
58 echo "|> Starting domain \"`$DOMAIN`\""
59 virsh start --domain $DOMAIN
```

Le fichier preseed.cfg.in :

© Contenu masqué n°4

6. Conclusion

5.1. Petit bonus : installation automatique d'une clé SSH

C'est un peu crade, mais l'installateur debian permet de lancer une commande en post-installation. Chez moi, ça rajoute la clé SSH que j'utilise partout :

```
1 d-i preseed/late_command string in-target sed -i '/cdrom/d'
   /etc/apt/sources.list; chroot /target sh -c "mkdir -pm 700
   /root/.ssh && echo 'ssh-ed25519 AAAAC3Nza...US4uhq553XgYq8ltBP6
   root@tac' > /root/.ssh/authorized_keys && chmod 0600
   /root/.ssh/authorized_keys && chown -R root:root /root/.ssh"
```

6. Conclusion

J'espère que ce premier billet vous a plu. J'écris ça à l'arrache en partie dans le train ; n'hésitez pas à me le signaler si vous trouvez des coquilles, ou si je ne suis pas clair.

Quelques liens en vrac :

https://raymii.org/s/articles/virt-install_introduction_and_copy_paste_distro_install_commands.html ↗

<https://www.debian.org/releases/stable/amd64/apb.html.en> ↗

<http://www.debian.org/releases/stretch/example-preseed.txt> ↗

Contenu masqué

Contenu masqué n°1

Exemple de commande QEMU que libvirt lance pour moi :

```
1 qemu-system-x86_64 -enable-kvm -name
   guest=devstack,debug-threads=on -S
2 -object
   secret,id=masterKey0,format=raw,file=/var/lib/libvirt/qemu/domain-5-dev
3 -machine pc-i440fx-2.8,accel=kvm,usb=off,dump-guest-core=off
   -cpu Opteron_G3 -m 2048
4 -realtime mlock=off -smp 4,sockets=4,cores=1,threads=1 -uuid
   6a938435-d5e3-4d01-95e7-4a2a250b0b95
5 -display none -no-user-config -nodefaults
6 -chardev
   socket,id=charmonitor,path=/var/lib/libvirt/qemu/domain-5-devstack/mon
7 -mon chardev=charmonitor,id=monitor,mode=control -rtc
   base=utc,driftfix=slew
```

```
8 -global kvm-pit.lost_tick_policy=delay -no-hpet -no-shutdown
   -global PIIX4_PM.disable_s3=1
9 -global PIIX4_PM.disable_s4=1 -boot strict=on
10 -device ich9-usb-ehci1,id=usb,bus=pci.0,addr=0x3.0x7
11 -device ich9-
   usb-uhci1,masterbus=usb.0,firstport=0,bus=pci.0,multifunction=on,addr=
12 -device ich9-
   usb-uhci2,masterbus=usb.0,firstport=2,bus=pci.0,addr=0x3.0x1
13 -device ich9-
   usb-uhci3,masterbus=usb.0,firstport=4,bus=pci.0,addr=0x3.0x2
14 -device virtio-serial-pci,id=virtio-serial0,bus=pci.0,addr=0x4
15 -drive
   file=/var/lib/libvirt/images/devstack.qcow2,format=qcow2,if=none,id=dr
16 -device
   virtio-blk-pci,scsi=off,bus=pci.0,addr=0x5,drive=drive-virtio-disk0,id=
   -netdev tap,fd=28,id=hostnet0,vhost=on,vhostfd=30
17 -device
   virtio-net-pci,netdev=hostnet0,id=net0,mac=52:54:00:cd:8a:3f,bus=pci.0
18 -chardev pty,id=charserial0 -device
   isa-serial,chardev=charserial0,id=serial0
19 -chardev
   socket,id=charchannel0,path=/var/lib/libvirt/qemu/channel/target/doma
20 -device
   virtserialport,bus=virtio-serial0.0,nr=1,chardev=charchannel0,id=channe
21 -device usb-tablet,id=input0,bus=usb.0,port=1
22 -device virtio-balloon-pci,id=balloon0,bus=pci.0,addr=0x6 -msg
   timestamp=on
```

[Retourner au texte.](#)

Contenu masqué n°2

```
1 gateway# ifconfig -A
2 lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 32768
3 ...
4 vio0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
5 lladdr 52:54:00:11:6e:e9
6 index 1 priority 0 llprio 3
7 groups: egress
8 media: Ethernet autoselect
9 status: active
10 inet 130.79.49.198 netmask 0xffffffc0 broadcast
   130.79.49.255
11 inet 130.79.49.208 netmask 0xffffffff
12 inet 130.79.49.209 netmask 0xffffffff
13 ...
```

```
14         inet 130.79.49.214 netmask 0xffffffff
15         inet 130.79.49.215 netmask 0xffffffff
16         ...
17 vio1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
18         lladdr 52:54:00:d7:77:65
19         index 2 priority 0 llprio 3
20         media: Ethernet autoselect
21         status: active
22         inet 10.0.0.1 netmask 0xffff0000 broadcast 10.0.255.255
23         ...
```

Extrait de `/etc/pf.conf` :

[Retourner au texte.](#)

Contenu masqué n°3

Le fichier de preseed complet que j'utilise. La clé `HOSTNAME` et `PASSWORD` sont remplacés à la volée par un moteur de template.

```
1 d-i debian-installer/locale select en_US
2 d-i console-setup/ask_detect boolean false
3 d-i keyboard-configuration/xkb-keymap select fr(latin9)
4
5 d-i netcfg/choose_interface select auto
6 d-i netcfg/get_hostname string HOSTNAME
7 d-i netcfg/get_domain string int.aius.u-strasbg.fr
8 d-i netcfg/disable_dhcp boolean false
9 d-i mirror/country string manual
10 d-i mirror/http/hostname string ftp.fr.debian.org
11 d-i mirror/http/directory string /debian
12 d-i mirror/http/proxy string
13 d-i netcfg/wireless_wep string
14
15 d-i clock-setup/utc boolean true
16 d-i time/zone select Europe/Paris
17
18 d-i partman-auto/disk string /dev/vda
19 d-i partman-auto/method string lvm
20 d-i partman-auto/purge_lvm_from_device boolean true
21 d-i partman-lvm/device_remove_lvm boolean true
22 d-i partman-lvm/device_remove_lvm_span boolean true
23 d-i partman-lvm/confirm boolean true
24 d-i partman-auto/choose_recipe select atomic
```

```
25
26 d-i partman/confirm_write_new_label boolean true
27 d-i partman/choose_partition select finish
28 d-i partman/confirm boolean true
29 d-i partman/confirm_nooverwrite boolean true
30
31 d-i passwd/root-login boolean true
32 d-i passwd/make-user boolean false
33 d-i passwd/root-password password PASSWORD
34 d-i passwd/root-password-again password PASSWORD
35
36 tasksel tasksel/first multiselect standard ssh-server
37
38 # Workaround for
    https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=666974
39 d-i grub-installer/only_debian boolean false
40 d-i grub-installer/bootdev string /dev/vda
41
42 # d-i apt-setup/non-free boolean true
43 # d-i apt-setup/contrib boolean true
44 d-i apt-setup/use_mirror boolean true
45
46 d-i apt-setup/security_host string security.debian.org
47 d-i apt-setup/services-select multiselect security, updates
48
49 d-i debian-installer/add-kernel-opts string console=ttyS0,115200n8
    serial
50
51 popularity-contest popularity-contest/participate boolean false
52
53 d-i finish-install/reboot_in_progress note
```

[Retourner au texte.](#)

Contenu masqué n°4

```
1 d-i debian-installer/locale select en_US
2 d-i console-setup/ask_detect boolean false
3 d-i keyboard-configuration/xkb-keymap select fr(latin9)
4
5 d-i netcfg/choose_interface select auto
6 d-i netcfg/get_hostname string HOSTNAME
7 d-i netcfg/get_domain string int.aius.u-strasbg.fr
8 d-i netcfg/disable_dhcp boolean false
9 d-i mirror/country string manual
10 d-i mirror/http/hostname string ftp.fr.debian.org
```

```
11 d-i mirror/http/directory string /debian
12 d-i mirror/http/proxy string
13 d-i netcfg/wireless_wep string
14
15 d-i clock-setup/utc boolean true
16 d-i time/zone select Europe/Paris
17
18 d-i partman-auto/disk string /dev/vda
19 d-i partman-auto/method string lvm
20 d-i partman-auto/purge_lvm_from_device boolean true
21 d-i partman-lvm/device_remove_lvm boolean true
22 d-i partman-lvm/device_remove_lvm_span boolean true
23 d-i partman-lvm/confirm boolean true
24 d-i partman-auto/choose_recipe select atomic
25
26 d-i partman/confirm_write_new_label boolean true
27 d-i partman/choose_partition select finish
28 d-i partman/confirm boolean true
29 d-i partman/confirm_nooverwrite boolean true
30
31 d-i passwd/root-login boolean true
32 d-i passwd/make-user boolean false
33 d-i passwd/root-password password pas-le-vrai
34 d-i passwd/root-password-again password pas-le-vrai
35
36 tasksel tasksel/first multiselect standard ssh-server
37
38 # Workaround for
    https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=666974
39 d-i grub-installer/only_debian boolean false
40 d-i grub-installer/bootdev string /dev/vda
41
42 # d-i apt-setup/non-free boolean true
43 # d-i apt-setup/contrib boolean true
44 d-i apt-setup/use_mirror boolean true
45
46 d-i apt-setup/security_host string security.debian.org
47 d-i apt-setup/services-select multiselect security, updates
48
49 d-i debian-installer/add-kernel-opts string console=ttyS0,115200n8
    serial
50
51 popularity-contest popularity-contest/participate boolean false
52
53 d-i finish-install/reboot_in_progress note
```

[Retourner au texte.](#)

Liste des abréviations

DI Direction Informatique. 3