

[Aller au contenu](#)

I. Installation de KVM

► Après avoir activé la virtualisation dans le BIOS, nous avons installé les paquets suivants :

```
1 • qemu-kvm : émule les machines virtuelles
2 • libvirt : librairie qui gère la virtualisation
3 • libvirt-clients
4 • libvirt-daemon-system
5 • virtinst : jeu de commande permettant la gestion des Vms
6 • bridge-utils : permet de créer le pont
7
8 apt-get install XXX
```

“XXX” est à remplacer par le nom du paquet souhaité.

```
1 apt-get install cpu-checker
2 (puis) kvm-ok (pour vérifier si la virtualisation est activée)
```

Création d'un bridge

► Dans le rc.local :

```
1 ifconfig enp4s0 up 0.0.0.0
2 brctl addbr br1
3 ifconfig br1 10.31.240.1/25
4 route add default gw 10.31.240.126
5 brctl addif br1 enp4s0
```

► Une fois fait, il suffit d'enregistrer puis de faire “./rc.local”. Il faut faire 2 bridge car il y a 2 réseaux, un privé et un public.

► Il a fallu faire un deuxième bridge, or en faire un deuxième a fait bug le tout car le premier avait déjà été créé et donc cela a créé un doublon.

- Une fois le problème réglé, création de l'utilisateur std dans le groupe libvirt :

```
root@afrique-srv:~# adduser std libvirt
Ajout de l'utilisateur « std » au groupe « libvirt »...
Adding user std to group libvirt
Fait.
```

- Afin de vérifier que la création du bridge n'a pas créé de dysfonctionnement nous avons ping google pour vérifier qu'il y a encore internet sur le serveur. La connexion en ssh fonctionne aussi car j'ai prit le screen à partir de mon ordinateur portable.

```
root@afrique-srv:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=114 time=13.3 ms
^C
--- 8.8.8.8 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 13.290/13.290/13.290/0.000 ms
root@afrique-srv:~# ping google.fr
PING google.fr (142.250.200.227) 56(84) bytes of data.
64 bytes from mrs08s18-in-f3.1e100.net (142.250.200.227): icmp_seq=1 ttl=114 time=12.5 ms
64 bytes from mrs08s18-in-f3.1e100.net (142.250.200.227): icmp_seq=2 ttl=114 time=12.5 ms
```

Installation de virt-manager sur l'hôte

- Virt manager est interface graphique permettant de gérer les machines virtuelles sous KVM. La commande suivante permet de le télécharger.

```
1 | apt install virt-manager
```

```
1 | systemctl status libvirtd.service
```

- Cette commande permet de voir si le démon libvirtd est lancé et activé au démarrage.

```

root@afrique-srv:~# systemctl status libvirtd.service
● libvirtd.service - Virtualization daemon
   Loaded: loaded (/lib/systemd/system/libvirtd.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2021-09-14 13:48:30 CEST; 6 days ago
   TriggeredBy: ● libvirtd-ro.socket
                 ● libvirtd-admin.socket
                 ● libvirtd.socket
   Docs: man:libvirtd(8)
          https://libvirt.org
  Main PID: 433 (libvirtd)
    Tasks: 22 (limit: 32768)
   Memory: 3.7G
     CPU: 4min 39.478s
    CGroup: /system.slice/libvirtd.service
            └─433 /usr/sbin/libvirtd

sept. 14 13:48:21 afrique-srv systemd[1]: Starting Virtualization daemon...
sept. 14 13:48:30 afrique-srv systemd[1]: Started Virtualization daemon.

```

- Nous pouvons voir sur le screen que le démon est bien lancé et activité au démarrage

Création d'une VM

- Nous avons dimensionner la VM via l'interface graphique et non en ligne de commande.

Ajout de l'utilisateur std au groupe libvirt

```
1 | adduser std libvirt
```

- Après avoir créer un utilisateur STD, on s'enlève de l'utilisateur root (sur le serveur) pour passer en "std" une fois fait, on tape : afin d'ouvrir l'interface graphique

virt manager

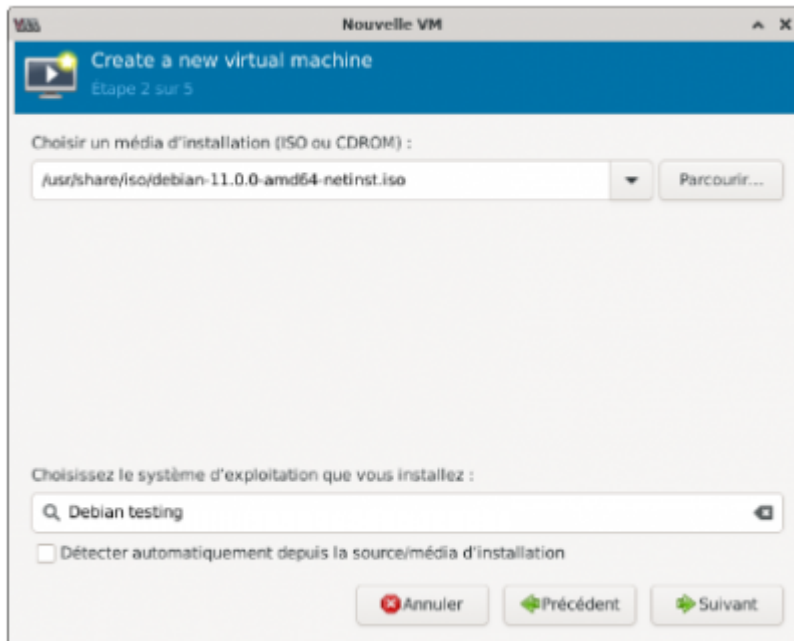
- Avant de commencer à créer une VM il a fallut modifier une ligne dans : /etc/libvirt/qemu.conf en ajoutant la directive :

```
security_driver="none"
```

- Afin de désactiver apparmor pour qemu.

- Une fois fait il faut redémarrer le démon `libvirtd.service` avec la commande :

```
1 | systemctl restart libvirtd.service
```



- Sélection des ressources pour la VM.

Nouvelle VM

Create a new virtual machine

Étape 3 sur 5

Choisir les paramètres mémoire et CPU :

Mémoire : 512 — +
Jusqu'à 3931 Mo disponibles sur l'hôte

CPU : 1 — +
Up to 3 available

Nouvelle VM

Create a new virtual machine

Étape 4 sur 5

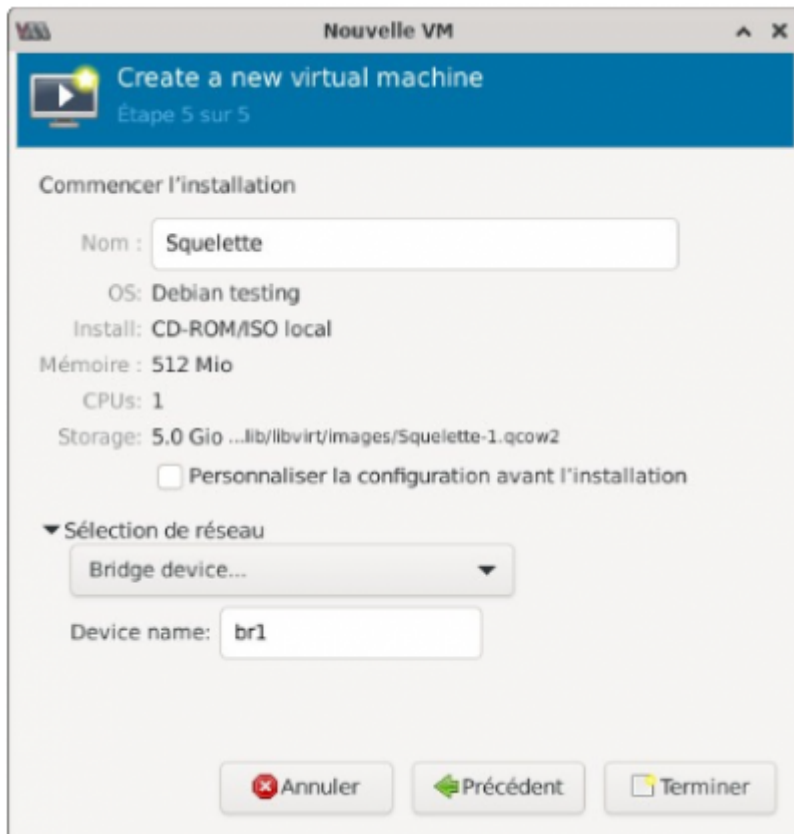
☒ Activer le stockage pour cette machine virtuelle

☒ Créer une image disque pour la machine virtuelle

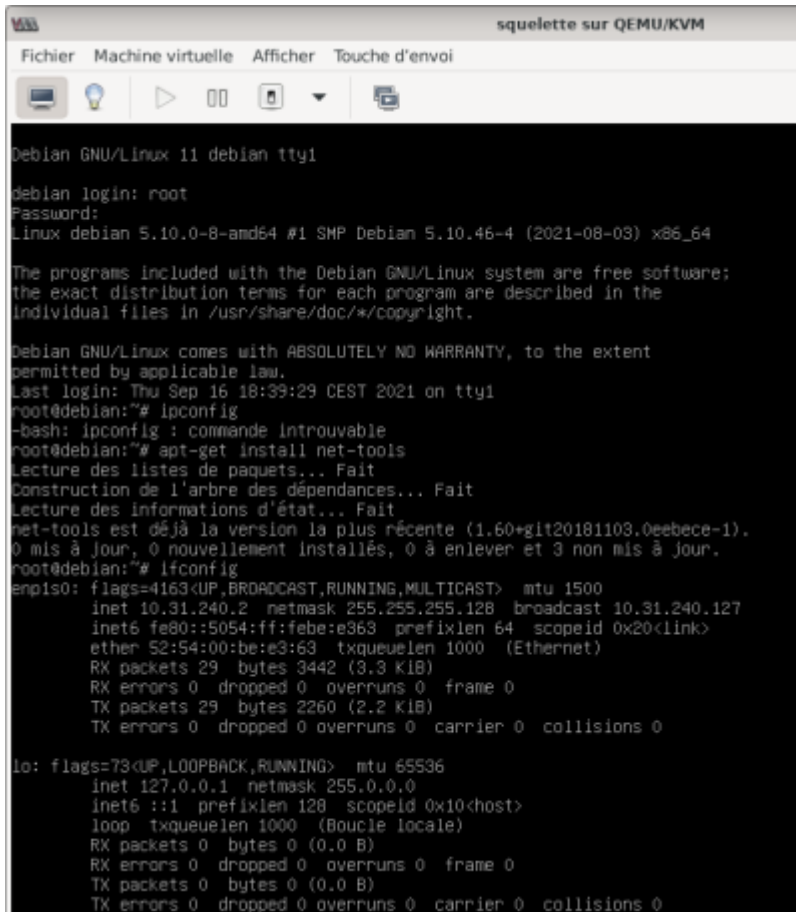
5,0 — + Gio
11.4 GiB disponible à l'emplacement par défaut

☐ Sélectionner ou créer un stockage personnalisé

Gérer...



- Pour l'installation de l'OS sur la VM se fait comme pour l'installation classique d'un OS.
- Pour cloner des VMs il faut être connecter en tant qu'std et taper "virt-manager".



```

squelette sur QEMU/KVM
Fichier  Machine virtuelle  Afficher  Touche d'envoi

Debian GNU/Linux 11 debian tty1
debian login: root
Password:
Linux debian 5.10.0-8-amd64 #1 SMP Debian 5.10.46-4 (2021-08-03) x86_64

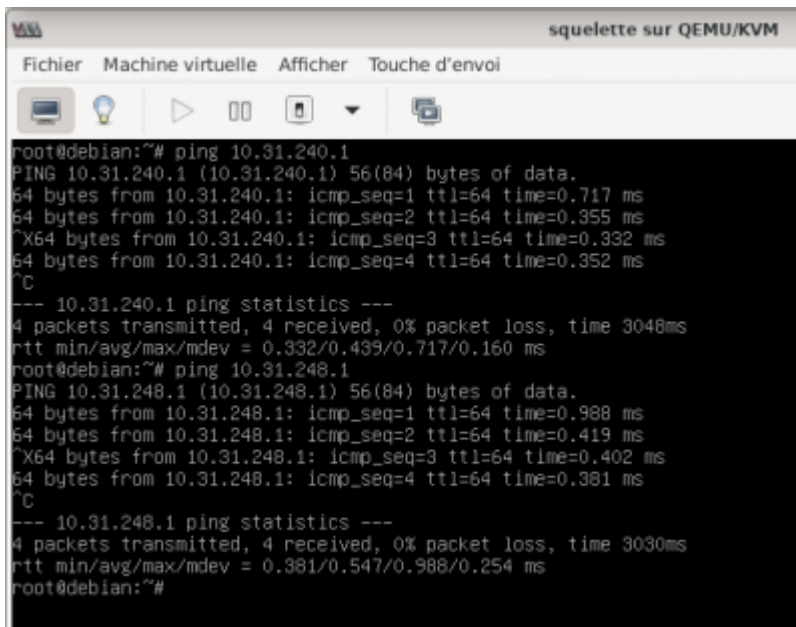
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Sep 16 18:39:29 CEST 2021 on tty1
root@debian:~# ipconfig
-bash: ipconfig : commande introuvable
root@debian:~# apt-get install net-tools
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
net-tools est déjà la version la plus récente (1.60+git20181103.0eebece-1).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 3 non mis à jour.
root@debian:~# ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.31.240.2  netmask 255.255.255.128  broadcast 10.31.240.127
    inet6 fe80::5054:ff:febe:e363  prefixlen 64  scopeid 0x20<link>
    ether 52:54:00:be:e3:63  txqueuelen 1000  (Ethernet)
    RX packets 29  bytes 3442 (3.3 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 29  bytes 2260 (2.2 KiB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
    loop txqueuelen 1000  (Boucle locale)
    RX packets 0  bytes 0 (0.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 0  bytes 0 (0.0 B)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

```

► Ici la configuration IP de notre VM squelette.



```

squelette sur QEMU/KVM
Fichier  Machine virtuelle  Afficher  Touche d'envoi

root@debian:~# ping 10.31.240.1
PING 10.31.240.1 (10.31.240.1) 56(84) bytes of data.
64 bytes from 10.31.240.1: icmp_seq=1 ttl=64 time=0.717 ms
64 bytes from 10.31.240.1: icmp_seq=2 ttl=64 time=0.355 ms
64 bytes from 10.31.240.1: icmp_seq=3 ttl=64 time=0.332 ms
64 bytes from 10.31.240.1: icmp_seq=4 ttl=64 time=0.352 ms
^C
--- 10.31.240.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3048ms
rtt min/avg/max/mdev = 0.332/0.439/0.717/0.160 ms
root@debian:~# ping 10.31.248.1
PING 10.31.248.1 (10.31.248.1) 56(84) bytes of data.
64 bytes from 10.31.248.1: icmp_seq=1 ttl=64 time=0.988 ms
64 bytes from 10.31.248.1: icmp_seq=2 ttl=64 time=0.419 ms
64 bytes from 10.31.248.1: icmp_seq=3 ttl=64 time=0.402 ms
64 bytes from 10.31.248.1: icmp_seq=4 ttl=64 time=0.381 ms
^C
--- 10.31.248.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3030ms
rtt min/avg/max/mdev = 0.381/0.547/0.988/0.254 ms
root@debian:~#

```

► On peut voir ici que les test ping des réseaux publique / privé fonctionnent parfaitement.

- Pour visualiser les paquets installés il faut taper la commande :

```
1 apt list --installed
```

```
perl-base/stable,now 5.32.1-4 amd64 [installé, pouvant être mis à jour vers : 5.32.1-4+deb11u1]
procps/stable,now 2:3.3.17-5 amd64 [installé]
python3-api-common/stable,now 2.2.1 all [installé, automatique]
python3-api/stable,now 2.2.1 amd64 [installé, automatique]
python3-certifi/stable,now 2020.6.20-1 all [installé, automatique]
python3-chardet/stable,now 4.0.0-1 all [installé, automatique]
python3-debian/stable,now 0.1.39 all [installé, automatique]
python3-debianopts/stable,now 3.1.0 all [installé, automatique]
python3-httplib2/stable,now 0.18.1-3 all [installé, automatique]
python3-idna/stable,now 2.10-1 all [installé, automatique]
python3-minimal/stable,now 3.9.2-3 amd64 [installé, automatique]
python3-pkg-resources/stable,now 52.0.0-4 all [installé, automatique]
python3-pycurl/stable,now 7.43.0.6-5 amd64 [installé, automatique]
python3-pysimpleasp/stable,now 1.16.2-3 all [installé, automatique]
python3-reportbug/stable,now 7.10.3 all [installé]
python3-requests/stable,now 2.25.1+dfsg-2 all [installé, automatique]
python3-six/stable,now 1.16.0-2 all [installé, automatique]
python3-unilib3/stable,now 1.26.5-1~exp1 all [installé, automatique]
python3.9-minimal/stable,now 3.9.2-1 amd64 [installé, automatique]
python3.9/stable,now 3.9.2-1 amd64 [installé, automatique]
python3/stable,now 3.9.2-3 amd64 [installé, automatique]
readline-common/stable,now 8.1-1 all [installé]
reportbug/stable,now 7.10.3 all [installé]
rsyslog/stable,now 8.2102.0-2 amd64 [installé]
runit-helper/stable,now 2.10.3 all [installé, automatique]
sed/stable,now 4.7-1 amd64 [installé]
sensible-utils/stable,now 0.0.14 all [installé]
ssh/stable,now 1:8.4p1-5 all [installé]
systemd-sysv/stable,now 247.3-6 amd64 [installé]
systemd-timesyncd/stable,now 247.3-6 amd64 [installé]
systemd/stable,now 247.3-6 amd64 [installé]
sysvinit-utils/stable,now 2.96-7 amd64 [installé]
tar/stable,now 1.34+dfsg-1 amd64 [installé]
taskset-data/stable,now 3.68 all [installé]
taskset/stable,now 3.68 all [installé]
tzdata/stable,now 2021a-1 all [installé]
ucf/stable,now 3.0043 all [installé]
udev/stable,now 247.3-6 amd64 [installé]
usbutils/stable,now 1:013-3 amd64 [installé]
util-linux/stable,now 2.36.1-8 amd64 [installé]
```

- Ici une petite partie de tous les paquets installés sur la VM squelette.