


# Qemu

- Objet : Qemu
- Niveau requis : [avisé](#)
- Commentaires : *logiciel libre de machine virtuelle*
- Suivi : [à-tester](#)
  - Création par  [captfnfab](#) le 17/08/2009
  - Testé par .... le ....
- Commentaires sur le forum : [C'est ici<sup>1\)</sup>](#)

## Résumé

Ce logiciel permet d'émuler un ou plusieurs ordinateurs sur votre propre machine, vous permettant par exemple d'installer une Debian pour tester la future version, ou d'installer un autre système d'exploitation :

- distribution Linux,
- os à base de noyau BSD,
- système propriétaire.
- ...

Il permet même d'émuler plusieurs architecture (arm, cris, i386, m68k, mips, mips64, ppc, ppc64, x86\_64, et d'autres !)

## Installation

Classiquement, on commence par mettre à jour la liste des paquets :

```
apt-get update
```

puis on installe qemu :

```
apt-get install qemu qemu-system-x86 qemu-utils
```

## Configuration

### Répertoire de travail

On commence par se créer un répertoire de travail, par exemple : ~/vm

```
mkdir -p ~/vm
```

# Création de machine virtuelle

```
cd ~/vm
```

Pour faire plus propre, on va maintenant créer un dossier pour la machine en question.

```
mkdir fenetresXP
```

```
cd fenetresXP
```

## L'image disque

On crée une image disque<sup>2)</sup> (de 20 Gio<sup>3)</sup> dans notre exemple)

```
qemu-img create -f qcow2 base.qcow2 20G
```

## Les périphériques

Ils sont choisis au lancement de qemu, par le biais d'options données en ligne de commande

### Pour choisir la carte son

```
-soundhw bla
```

Où bla peut prendre une des valeurs renvoyées par `qemu -soundhw ?`

### Pour changer de carte graphique

```
-vga bla
```

Où bla peut prendre une valeur parmi « cirrus, std, vmware, none »

### Pour l'usb

- Active le support de l'usb

```
-usb
```

- Et pour ajouter les périphériques USB :

```
-usbdevice bla
```

*bla* peut être remplacé par

1. `host:bus.addr` pour affecter un port usb (
2. `host:vendor_id:product_id` pour affecter un périphérique

Exemple :

Si la commande :

```
lsusb
```

Me renvoie :

[retour de la commande](#)

```
Bus 007 Device 003: ID 0b97:7772 Blahhhhhhhhhh  
Bus 007 Device 002: ID 0b97:7761 Ploum ploum
```

Si je veux monter le port USB sur lequel est branché mon périphérique Blahhhhhhhhhh

```
-usbdevice "host:007.003"
```

Si je veux monter Blahhhhhhhhhh où qu'il soit branché

```
-usbdevice "host:0b97:7772"
```

### Nombre de microprocesseurs à allouer à la machine

```
-cpu n
```

(où n représente ce nombre)

### Taille de la mémoire vive à allouer à la machine

```
-m n
```

(où n est cette taille en Mio)

### Et bien d'autres options, ...

... notamment pour le réseau (présent par défaut) Voir man ou documentation en référence.

### Enregistrement des options

Il est fastidieux de se souvenir des options à chaque lancement, on va donc les enregistrer dans un fichier :

```
touch "launch_vm.sh"
```

On édite le fichier de sorte à ce qu'il ressemble à ça :

[launch\\_vm.sh](#)

```
#!/bin/sh
qemu -m 256 -soundhw all -vga cirrus -hda ~/vm/fenetresXP/base.qcow2 $*
```

On le rend exécutable :

```
chmod a+x "launch_vm.sh"
```

## Lancement de la machine la première fois

On veut booter sur le CDRom d'installation, donc

```
./launch_vm.sh -m 512 -cdrom ~/mes_isos/fenetresXP_SP3.iso -boot d
```

## Lancement de la machine les autres fois

```
./launch_vm.sh bla
```

où *bla* sont les options que vous voulez rajouter.

## Aller plus loin : faire des tests sans pourrir sa config

Si vous utilisez votre vm pour tester des logiciels non stable ou des virus, vous n'avez certainement pas envie de ~~foutre en l'air, détruire~~, perdre votre jolie installation vierge toute belle toute neuve sur votre machine virtuelle.

Faire une copie de sauvegarde de l'image prendrait beaucoup de place.

Heureusement, il y a mieux !

Supposons que vous veniez de suivre l'installation ci-dessus. Vous avez installé fenêtre XP SP3, effectué plein de mises à jour, installé un navigateur web, .PasNet Framework et autres machins lourds... Et vous estimez que cette configuration est le point de départ idéal pour tous vos tests futurs.

On s'assure d'être dans le dossier de la vm :

```
cd ~/vm/fenetresXP
```

On interdit les droits d'écriture sur l'image de départ

```
chmod 444 base.qcow2
```

Et on crée une image test1.qcow2 qui ne contiendra que les modifications faites à base.qcow2

```
qemu-img create -b base.qcow2 -f qcow2 test1.qcow2
```

Et voilà, maintenant, il suffit de modifier le script launch\_vm.sh pour qu'il utilise test1.qcow2 au lieu de base.qcow2

### Astuce

Avec le script suivant launch\_vm2.sh :

[script launch\\_vm2.sh](#)

```
#!/bin/bash
IMG=$1
shift;
qemu -m 256 -soundhw all -vga cirrus -hda ~/vm/fenetresXP/$IMG.qcow2 $*
```

Vous lancez la machine de base (attention, en lecture seule) avec :

```
./launch_vm2.sh base
```

La machine de test 1 :

```
./launch_vm2.sh test1
```

Etc. Vous pouvez simultanément avoir 12<sup>4)</sup> machines de test différentes basées sur la même image base.qcow2, et ce, pour un coût en espace disque moindre !

## Quelques raccourcis indispensables

<b>RACCOURCIS CLAVIER</b>	<b>ACTION</b>
<b>Ctrl+Alt+F</b>	passage en plein-écran
<b>Ctrl+Alt+1</b>	affichage graphique de l'invité
<b>Ctrl+Alt+2</b>	passage au moniteur Qemu; attention, dans ces deux options, le 1 et le 2 sont à taper sur les chiffres hauts du clavier, pas sur le pavé numérique
<b>Ctrl+Alt</b>	pour arrêter le contrôle de la souris dans l'invité, alors que pour l'activer, on avait cliqué dans la fenêtre de Qemu

voir le très bon tuto sur lea-linux là :

[http://lea-linux.org/documentations/index.php/Software-soft\\_emul-qemu](http://lea-linux.org/documentations/index.php/Software-soft_emul-qemu)

Lien : Voir : [KVM - Fork de Qemu](#)

## Références

1. man qemu (anglais)
2. <http://www.esaracco.fr/documentation/qemu/qemu/> (ou en pdf) sur le site <http://www.esaracco.fr/>

1)

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

2)

le format qcow2 nous permet de ne pas créer directement un fichier de 20 Gio mais de n'allouer l'espace que lorsqu'il est requis

3)

Pour le reste du tuto, on considèrera que  $1 \text{ Gio} = 2^{10} \text{ Mio} = 2^{20} \text{ kio} = 2^{30} \text{ o}$

4)

pour toute valeur de 12

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/doc:systeme:vm:qemu>

Last update: **19/11/2020 14:09**

