

TP3 - Utilisation du logiciel de virtualisation KVM/QEMU

*L'objectif de ce TP est d'étudier comment déployer une machine virtuelle de type KVM. KVM, pour Kernel Virtual Machine, est un hyperviseur libre de type "1 / 2" pour Linux. KVM est intégré dans le noyau Linux depuis la version 2.6.20, soit approximativement depuis 2007. C'est un hyperviseur plutôt populaire, il est par exemple l'hyperviseur préféré d'[OpenStack](#) et est donc largement utilisé dans le monde du Cloud. **IMPORTANT : se connecter avec le compte tpreseau, puis changer le mot de passe de root et de tpreseau sur la machine hôte (la "vraie") !!**. Les manipulations à venir sont à faire depuis le compte tpreseau si les commandes sont précédées de \$ et depuis le compte super-utilisateur si elles débutent par #.*

1 Introduction

Comme KVM est basé sur Linux, ce n'est pas pleinement un hyperviseur de type 1, ou bare-metal. On peut l'installer via des commandes sur n'importe quel système Linux que l'on peut alors utiliser comme serveur de virtualisation. De fait, KVM peut aussi être vu comme un hyperviseur de type 2, ce qui signifie qu'il s'exécute dans le niveau non-privilegié du *x86 Privilege Ring*, soit le niveau *ring 3*. Le fait que KVM soit un hyperviseur de type 1 ou 2 est l'objet d'une discussion déjà très ancienne. Bien entendu, le système sur lequel on veut utiliser KVM doit supporter une des deux extensions Intel VT-x (*Virtualization Technology extension*) ou AMD-V (*AMD64 Virtualization extension*).

2 Préparation / installation de KVM

Avant de mettre en place KVM, on va vérifier pour la forme que la machine hôte supporte la virtualisation, puis on mettra à jour le système et installera les paquets nécessaires.

Les manipulations à effectuer sont :

1. vérifier que le processeur dispose des flags `vmx` ou `svm` ;

```
# egrep -c '^flags.* (vmx|svm)' /proc/cpuinfo
```

normalement cela doit vous donner une valeur plus grande que 0. Vous pouvez aussi produire un affichage des flags en couleur avec

```
# grep -E --color '^flags.* (vmx|svm)' /proc/cpuinfo
```

2. mettre le système à jour ;
3. on installe QEMU-KVM et libvirt, mais avant cela consulter le site de [libvirt](#)

```
# apt install --no-install-recommends qemu-system libvirt-clients \
libvirt-daemon-system
```

L'option `--no-install-recommends` vise à empêcher l'installation de certains paquets annexes ;

ATTENTION : vous pouvez consulter également le [Wiki Debian](#) dont une partie des étapes qui suivent sont issues.

4. vérifier avec `lsmod` que l'un des modules `kvm_intel` ou `kvm_amd` est bien chargé ;
5. afin de pouvoir créer des machines depuis la ligne de commande on installe `virtinst`, ainsi que `dnsmasq`

```
# apt install virtinst dnsmasq
```

6. on ajoute l'utilisateur `tpreseau` aux groupes `kvm` et `libvirt`, soit avec la commande `adduser`, soit avec `usermod` en étant `root` ;
7. on peut regarder les réseaux disponibles, pour cela vous lancerez également cette commande en étant `root`

```
# virsh net-list --all
```

puis en étant `tpreseau`, auquel cas vous devriez voir une différence ;

8. vous devriez avoir constaté au niveau de l'affichage obtenu pour `root` que le réseau par défaut n'est pas actif et ne démarre pas automatiquement au boot. Nous allons donc faire ce qu'il faut pour qu'il soit actif et qu'il démarre tout seul

```
# virsh net-start default
# virsh net-autostart default
```

9. pour améliorer les performances, on charge avec `modprobe` le module `vhost_net`, pour plus d'information vous pouvez consulter le lien [suivant](#) ;
10. l'utilisateur peut consulter la liste des machines virtuelles via

```
$ virsh list --all
```

11. cependant, par défaut seules les machines appartenant à l'utilisateur seront visibles car la connexion à `libvirt` va se faire via `qemu:///session`. Si l'utilisateur veut pouvoir gérer les VM du *system set*, il faut se connecter comme suit

```
$ virsh --connect qemu:///system list --all
```

pour éviter d'avoir à ajouter l'option `--connect` à chaque commande, il est possible de définir la variable d'environnement `LIBVIRT_DEFAULT_URI` avec

```
export LIBVIRT_DEFAULT_URI='qemu:///system'
```

Naturellement, cette définition n'est valable que pour la session en cours. Si vous voulez qu'elle devienne pérenne, une solution est d'ajouter cette définition dans le `.bashrc` de l'utilisateur. On vous demande donc d'ajouter la définition à la fin du fichier `.bashrc` dans le compte `tpreseau`, puis de rebooter la machine et de vérifier que la variable est bien définie avec la commande `env` dans un terminal de `tpreseau`. On vérifie également que l'utilisateur voit cette fois bien l'interface réseau via

```
$ virsh
virsh # net-list --all
virsh # help
virsh # quit
```

3 Création de machines virtuelles

Pour créer une machine, il est possible d'utiliser la ligne de commande grâce à *virtinst* comme on le verra ci-dessous et bien entendu, il existe également des applications graphiques telles que :

- AEQMU (*aqemu*);
- Virtual Machine Manager (*virt-manager*).

3.1 Création et gestion via la ligne de commande

Pour commencer nous allons voir comment installer une VM à partir d'exemples (issus du Wiki) permettant de créer une machine Debian avec la distribution *Bullseye*. Puis, en vous basant sur ces informations, vous devrez installer une machine virtuelle avec Debian et une autre avec la dernière version 18.04 Bionic Beaver d'Ubuntu. La machine Debian sera installée avec *virt-install*, tandis que Ubuntu le sera via *qemu-img* et *kvm* (ou *qemu-system-x86_64*, les deux sont équivalents).

3.1.1 Création de vm-passerelle - VM Debian 10

Voici des exemples de création avec *virt-install*.

- Un premier exemple de création de la machine *bullseye-amd4* qui suppose que l'image ISO a été téléchargée.

```
virt-install --virt-type kvm --name bullseye-amd64 \
--cdrom ~/iso/Debian/debian-11.0.0-amd64-netinst.iso \
--os-variant debian11 \
--disk size=10 --memory 1000
```
- Dans le cas où le système invité n'a pas d'accès réseau fonctionnel, il faut utiliser *virt-viewer* pour terminer l'installation.
- Dans ce second exemple on donne directement URL de l'ISO avec l'option *-location*

```
virt-install --virt-type kvm --name bullseye-amd64 \
--location http://deb.debian.org/debian/dists/bullseye/main/installer-amd64/ \
--os-variant debian11 \
--disk size=10 --memory 1000
```
- Ce dernier exemple va faire une installation en mode texte via la console *ttyS0*

```
virt-install --virt-type kvm --name bullseye-amd64 \
--location http://deb.debian.org/debian/dists/bullseye/main/installer-amd64/ \
--os-variant debian11 \
--disk size=10 --memory 1000 \
--graphics none \
--console pty,target_type=serial \
--extra-args "console=ttyS0"
```

Les manipulations à effectuer sont :

1. on installe le paquet *libosinfo-bin* puis on affiche la liste des OS

```
# apt install libosinfo-bin
$ osinfo-query os | more
```
2. on va placer les machines dans le compte utilisateur via un lien comme suit, il faut passer *root* après le *mkdir*

```
$ mkdir -p ~/tpreseau/Qemu-KVM\ VMs
# cd /var/lib/libvirt
# rmdir images
# ln -s /home/tpreseau/Qemu-KVM\ VMs images
```

3. lancer la commande `ip` a pour voir l'interface `virbr0` qui a été créé lors de l'installation de `virtinst` et qui sera donc utilisée pour permettre aux VMs d'accéder à Internet ou télécharger des fichiers lors de l'installation de la VM ;
4. à partir du dernier exemple ci-dessus créer en ligne de commande une machine virtuelle avec les caractéristiques suivantes :
 - comme nom mettre `vm-passerelle` ;
 - le système d'exploitation sera de type `Linux`, avec `Debian 10 Buster` (64 bits) pour la version / variante ;
 - pour la mémoire mettre 2 Gio ;
 - créer un disque dur virtuel de 12 Gio.
 Utiliser un partitionnement automatique et n'installer que le minimum, à savoir `SSH server` et `standard system utilities` ;
5. une fois la machine installée, s'y connecter avec le compte utilisateur créé et regarder sa configuration réseau ;
6. ouvrir un terminal et afficher la liste des VMs. Pouvez-vous la voir ? Si ce n'est pas le cas, faire ce qu'il faut pour que vous puissiez la voir ;
7. utiliser le Wiki pour arrêter la machine, puis la redémarrer ;

ATTENTION : il vous faudra éventuellement regarder sur d'autres sites que le Wiki pour obtenir des informations complémentaires.
8. pouvez-vous vous connecter depuis la VM dans la machine hôte et inversement ?
9. lancer `virsh`, le terminal de virtualisation interactif, dans un terminal ;
10. afficher ensuite l'aide générale avec `help`, puis une aide plus précise avec par exemple `virsh help shutdown` ;
11. modifier la configuration réseau par défaut pour que les adresses IP soient dans la même plage que dans *VirtualBox*.

3.1.2 Création de vm-texte - VM Debian 11

Pour cette machine, nous allons utiliser l'outil `virt-builder`, qui nécessitera d'installer un paquet supplémentaire, en plus de `virt-install`. Ce paquet permettra également de disposer de certaines commandes permettant d'aller dans le système de fichier d'une VM sans la lancer.

Les manipulations à effectuer sont :

1. on installe le paquet nécessaire

```
# apt -y install libguestfs-tools
```

2. on affiche la liste des OS qui sont disponibles

```
$ virt-builder -l | more
```

vous devriez constater que `debian-11` est disponible ;

3. on crée une image (un disque virtuel) d'une Debian 11 via le compte utilisateur avec

- ```
$ cd /var/lib/libvirt/images
$ virt-builder debian-11 --format qcow2 --size 10G -o vm-texte.qcow2 \
--hostname vm-texte --root-password password:tpreseau
```
- on utilise `virt-install` pour créer la VM avec l'image ci-dessus

```
$ virt-install --name vm-texte --memory 4096 \
--disk path=/var/lib/libvirt/images/vm-texte.qcow2 --vcpus 2 \
--os-type generic --network bridge=virbr0 --graphics none --console pty \
--boot hd --noreboot
```
  - démarrer la machine virtuelle, s'y connecter avec le compte `root` et regarder le nombre de paquets pouvant être mis à jour, puis se déconnecter ;
  - on arrête la VM et on la détruit

```
$ virsh shutdown vm-texte
$ virsh undefine vm-texte
$ virsh list --all
```

ensuite, on pourrait supprimer le fichier correspondant au disque virtuel, ce que l'on ne vas pas faire ;
  - on recrée la VM comme plus haut avec `virt-install` et on la lance, même si il y a des messages d'erreur. Normalement la machine démarre sans problème ;
  - mettre à jour la machine, ajouter le compte `tpvirtual` et vérifier que l'on peut se connecter à la VM depuis la VM `vm-passerelle` avec SSH ;
  - comme indiqué précédemment d'autres commandes permettent d'accéder au disque virtuel depuis un terminal, que la VM soit démarrée ou non. Voici quelques commandes à tester :
- ```
$ virt-ls -l -d vm-texte /root
$ virt-cat -d vm-texte /etc/passwd
$ virt-edit -d vm-texte /etc/fstab
```

Il y en a bien d'autres.

3.1.3 Construction de vm-graphique - VM Ubuntu 18.04

- Création de la machine virtuelle

Les manipulations à effectuer sont :

- créer le fichier correspondant au disque virtuel de la machine comme suit

```
$ cd /var/lib/libvirt/images
$ qemu-img create -f qcow2 vm-graphique.qcow2 10G
```

- récupérer l'image ISO de la version 18.04 d'Ubuntu sur l'URL

<http://archive.ubuntu.com/ubuntu/dists/bionic-updates/main/installer-amd64/current/images/netboot/mini.iso>

- lancez la création / l'installation avec la commande `kvm` ou bien `qemu-system-x86_64`.

Les options à utiliser sont :

- `-m` pour définir la taille de la mémoire en Mio → mettre 4 Gio ;
- `-cpu` pour définir le processeur → `host` ;
- `-cdrom` pour définir l'image ISO à utiliser pour faire l'installation ;

- `-boot d` pour définir le disque à booter en le mettant après, soit après l'option `-cdrom`;
 - on termine par le nom du fichier correspondant au disque virtuel à utiliser.
4. suite au message qui est affiché, installer l'outil qui vous permettra de vous connecter à la VM si il n'est pas déjà installé ;
 5. l'utiliser pour s'y connecter et procéder à l'installation en utilisant le disque en entier ;
 6. lors de l'installation, installer les logiciels associés à `Xubuntu minimal installation`.
 - Arrêt de la machine virtuelle

Nous allons voir comment accéder au moniteur Qemu qui permet de faire cela et bien plus.

Les manipulations à effectuer sont :

1. se connecter sur le site [Wiki Debian Facile](#) et plus précisément sur la partie dédiée à KVM avec votre navigateur préféré ;
2. regarder comment accéder au moniteur Qemu, puis via l'option `help`, trouver la commande permettant d'arrêter la machine ;
3. lancer la commande `virsh list --all`. Que constatez-vous ?
 - Lancement de la machine virtuelle

Les manipulations à effectuer sont :

1. se placer dans le répertoire `/var/lib/libvirt/images` ;
2. lancer la VM via


```
$ kvm -m 4G -cpu host -boot d vm-graphique.qcow2 &
```
3. s'y connecter avec le compte utilisateur créé et regarder sa configuration réseau. Qu'en déduisez-vous ? Lancer la VM `vm-texte`, les deux VM se voient-elles ?
4. installer la navigateur web `firefox` ;
5. arrêter la VM ;
6. la redémarrer, puis faire en sorte qu'elle ait accès à l'image ISO d'Ubuntu. Pour cela le Wiki devrait vous aider, sinon faites une recherche sur Internet avec les mots clés *change cdrom kvm qemu* ;
7. finalement arrêter la machine.

3.2 Création et gestion via une interface graphique

On commence par installer l'outil que nous allons utiliser, à savoir *Virt-Manager*.

Les manipulations à effectuer sont :

1. on installe l'outil via `apt`, il a été développé par RedHat pour créer, gérer et lancer des VMs créées avec KVM


```
# apt -y install virt-manager
```
2. lancer l'outil et vérifier que la VM Debian `vm-texte` est bien visible. Est-ce le cas de la machine Xubuntu `vm-graphique` ? Faire en sorte que cette dernière devienne visible en important l'image disque correspondante ;
3. les trois machines peuvent-elles se voir maintenant ?

Maintenant, on va utiliser l'outil pour créer une nouvelle machine à l'image de la machine `vm-graphique` créée lors d'un TP précédent.

Les manipulations à effectuer sont :

1. modifier la configuration de la machine Xubuntu `vm-graphique` via `Virt-Manager` pour que le nom de la machine devienne `vm-graphique-xubuntu`, le fichier `.qcow2` doit également être modifié en conséquence ;
2. démarrer la machine Xubuntu et vérifier qu'elle est fonctionnelle, modifier ensuite le nom de la machine (celui que l'on voit dans un terminal par exemple) pour qu'il devienne également `vm-graphique-xubuntu` ;
3. cliquer sur l'icône permettant de créer une nouvelle machine et en créer une que vous appellerez `vm-graphique-debian`. Compléter les informations en reprenant les caractéristiques de la machine `vm-graphique` données dans l'énoncé TP1. Vous utiliserez l'image ISO Debian que vous aurez téléchargé comme indiqué dans ce même énoncé. Vous laisserez l'image dans le répertoire Téléchargement de l'utilisateur `tpreseau`. Vous installerez MATE comme environnement de bureau Debian (en plus de serveur SSH et utilitaires usuels du système) ;
4. une fois l'installation terminée, lancer les machines `vm-texte` et les deux `vm-graphique`, peuvent-elles se voir ? En déduire la nature du réseau qui est mis en place dans la nouvelle machine `vm-graphique-debian`.

4 Configuration réseau

4.1 Le mode Accès par pont

Les manipulations à effectuer sont :

1. installer le package `bridge-utils`
2. on commence par regarder la configuration réseau de la machine hôte avec `ip a`, de façon à identifier le nom de l'interface réseau et l'adresse IP à utiliser, à savoir `eno1` et une adresse IP telle que `172.20.20.91` ;
3. on va ensuite modifier le fichier `interfaces`, en le modifiant comme suit à la fin

```
# The primary network interface
#allow-hotplug eno1
#iface eno1 inet dhcp

# The br0 bridge
auto br0
iface br0 inet dhcp
    bridge_ports eno1
```

4. ensuite on recharge la configuration réseau, en étant `root`, via
`/etc/init.d/networking restart`
5. faire en sorte que la machine `vm-graphique-debian` utilise le pont comme interface réseau ;
6. finalement vous mettrez en place `x11vnc` comme indiqué dans l'énoncé de TP1 et vous vérifierez que l'on peut prendre le contrôle de la machine.

4.2 Le mode Réseau NAT

En principe, vous avez constaté en utilisant les machines `vm-texte` et `vm-graphique-xubuntu` qu'elles sont dans un même réseau privé qui fonctionne en mode NAT. Pour finir, si d'aventure vous êtes arrivé jusque là, il s'agit de regarder comment contrôler la redirection de port de manière à pouvoir se connecter avec SSH sur la machine en mode texte via la machine hôte, comme nous l'avons vu lors des deux TP précédents.

À vous de jouer en utilisant les informations disponibles sur Internet pour faire cela.

5 Virtualisation avec bureau SPICE

SPICE, pour *Simple Protocol for Independent Computing Environments*, est un protocole d'affichage et d'interaction à distance dédié aux machines virtuelles. Pour plus d'information sur cette solution qui permet d'accéder à distance à des VMs, vous pouvez consulter ce [lien](#). SPICE est souvent utilisé de concert avec KVM/QEMU. Il est possible de démarrer une machine avec le support de SPICE en modifiant sa configuration, même si elle n'a pas été créée en l'activant.

Les manipulations à effectuer sont :

1. on commence par installer le client SPICE (il est possible qu'il soit déjà installé)

```
# apt -y install virt-viewer
```

2. on va ensuite éditer le fichier de configuration de la machine `vm-texte` et ajouter ce qu'il faut pour configurer un serveur SPICE. Tout d'abord, on ouvre avec un éditeur de texte le fichier `.xml` de la VM

```
virsh edit vm-texte
```

3. on modifie le fichier pour ajouter les deux sections ci-dessous juste avant la section `<memballoon>`

```
<graphics type='spice' port='5910' autoport='no' listen='0.0.0.0' passwd='CqriT'>
  <listen type='address' address='0.0.0.0' />
</graphics>
<video>
  <model type='qxl' ram='65536' vram='32768' heads='1' />
  <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0' />
</video>
```

4. on démarre la machine avec `virsh start vm-texte`;
5. on lance le client SPICE et on se connecte à la VM, la section `<graphics>` ci-dessus comporte des informations qui vous seront nécessaires ;
6. via le gestionnaire de machines virtuelles on regarde les détails de la VM et plus particulièrement l'**Affichage Spice** ;
7. on vous demande ensuite de mettre en place un accès SPICE sur la VM `vm-passerelle` en utilisant le gestionnaire de machines virtuelles, à savoir *Virt-Manager*, pour ajouter l'affichage SPICE et le configurer pour utiliser le port 5920 et le même mot de passe ;
8. démarrer `vm-passerelle` et vérifier que le client SPICE permet de s'y connecter ;
9. demander à un de vos camarades d'utiliser son client SPICE pour se connecter à votre VM `vm-passerelle` ;

10. vous pouvez essayer d'utiliser une tunnel SSH pour sécuriser les communications induites par la connexion SPICE entre votre machine et celle de votre camarade ;
11. au niveau de la configuration de `vm-graphique-xubuntu`, toujours avec le gestionnaire de machines virtuelles, modifier l'`Affichage spice` existant pour le transformer en `Affichage VNC`. Vous utiliserez le numéro de port 5930 et le même mot de passe que précédemment ;
12. lancer la VM `vm-graphique-xubuntu`, s'y connecter avec un client VNC. Que se passe-t-il au niveau de la console graphique de la VM dans le gestionnaire `Virt-Manager` ?

Il est possible de déjà définir une connexion SPICE lors de l'installation d'une VM et ainsi de procéder à une installation à distance via celle-ci. Normalement, vous avez vu précédemment, lors de l'installation des VMs, que c'était VNC qui permettait de faire cela. Dans le gestionnaire, il est possible de choisir SPICE ou VNC dans les préférences de `Nouvelle VM`. Voici un exemple de création d'une VM avec un accès SPICE :

```
virt-install \
--name Win2k19 \
--ram 6144 \
--disk path=/var/kvm/images/Win2k19.img,size=100 \
--vcpus=4 \
--os-type windows \
--os-variant=win2k19 \
--network bridge=br0 \
--graphics spice,listen=0.0.0.0,password=password,keymap=ja \
--video qxl \
--cdrom /tmp/Win2019_JA-JP.ISO
```

Questions subsidiaires pour les plus rapides

- Mettre des machines `VirtualBox` et `KVM/QEMU` dans un même réseau privé hôte et faire ce qu'il faut pour qu'elles puissent aller sur Internet.
 - Est-il possible de sécuriser l'accès SPICE directement avec TLS ?
 - Est-il possible de créer une VM dans une VM ?
- Ici on vous demande de regarder si c'est possible et pas nécessairement de le faire.