

TP1

Préparation de KVM

Il faut d'abord vérifier que votre machine supporte KVM. Ces commandes de préparation doivent se faire en tant que localuser.

Préparation de KVM :

1. Vérifiez que votre processeur dispose des flags vmx ou svm avec la commande `egrep '^flags.*(vmx|svm)' /proc/cpuinfo`.
 2. Installez les paquetages kvm et kvm-pxe (`sudo apt-get install qemu-kvm kvm libvirt-clients libvirt-daemon-system`).
 3. Vérifiez que le module `kvm_intel` ou `kvm_amd` est bien chargé (utilisez `sudo lsmod`). C'est ce module qui est chargé par défaut qui permet d'utiliser KVM.
 4. Créer deux groupes que vous nommerez `kvm` et `libvirt` avec la commande `groupadd`
 5. Ajoutez l'utilisateur local au groupe `kvm` avec la commande `adduser <user> kvm`
 6. Ajoutez l'utilisateur local au groupe `libvirt` avec la commande `adduser` ou faisant: `usermod -G libvirt <user>`
- aussi en
7. tapez `groups` pour vérifier que l'utilisateur local fait bien partie du groupe `kvm`.
 8. Créez un répertoire `/vservers` appartenant au groupe `kvm` et donnez les droits d'écriture au groupe `kvm` (vous utiliserez les commandes `mkdir`, `chgrp` ou `chown`). Vérifiez que la commande `ls -ld /vservers` ressemble bien à

```
drwxrwxr-x 3 root kvm 4096 2010-07-07 10:52 /vservers/
```

Création d'une machine virtuelle Windows

Nous allons maintenant installer un Windows 7 depuis Ubuntu ou Debian !

Création de la machine virtuelle :

1. Créez un fichier `/vservers/win7.qcow2` qui contiendra le disque dur de la machine virtuelle Windows avec `qemu-img create -f qcow2 /vservers/win7.qcow2 20G`
2. Comparez les résultats des commandes : `ls -lh /vservers/win7.qcow2`. Est-ce surprenant ?

3. Récupérez l'image ISO d'un CD windows, qui devrait circuler dans la salle sur une clé USB.

```
> mkdir /vservers/images
```

```
> cp /media/.../windows7.iso /vservers/images
```

Pour information, cette image ISO a été créée à partir d'un cd, avec la commande

```
> dd if=/dev/cdrom of=windows7.iso
```

4. Si vous êtes doué et partageur (!), exportez votre répertoire /vservers/images/ par NFS pour éviter à vos collègues d'attendre la clé.

5. Lancez l'installation de windows 7 avec

```
> kvm -m 2048 -cdrom /vservers/images/windows7.iso -boot d /vservers/win7.qcow2
```

Ou

```
> qemu-system-x86_64 -m 2048 -cdrom /vservers/images/windows7.iso -boot d /vservers/win7.qcow2
```

et profitez de cet instant d'installation pour comprendre les paramètres de la commande kvm que vous venez de lancer.

6. Choisissez formatage de tout le disque en ntfs rapide (la taille du disque dur était-elle prévisible ?).

7. Une fois l'installation terminée, arrêtez windows 7.

Lancement de windows 7

Nous allons lancer windows 7 et tester le réseau auquel il a accès.

1. Lancez win 7 avec `kvm -m 2048 /vservers/win7.qcow2` ou avec la commande `qemu-system-x86_64 -m 2048 /vservers/win7.qcow2`

2-Vérifiez que vous disposez d'un réseau (avec les commandes `ipconfig /all`, `ping`, . . .)

3-Que donne la commande `nslookup www.google.fr` ? Qu'est-ce que cela implique ?

4-Testez route PRINT ? Analysez la sortie.

5-Testez `ping 10.0.2.2`. Qui est 10.0.2.2 ?

6-Est ce que vous avez accès au web ?.

7-Par quel procédé technique windows 7 a-t-il accès au Web ?

8-Téléchargez et installez chrome ou Firefox.

9. Arrivez-vous à ping-er windows 7 depuis la machine hôte ?

10. Désactivez le pare-feu de windows 7, et réessayez. Est-ce que ça fonctionne ?

1.5 Création d'une machine virtuelle Linux

Nous allons maintenant installer une Debian (faites circuler l'image ISO) !

Création de la machine virtuelle :

1. Créez un fichier /vservers/debian.qcow2 qui contiendra le disque dur de la machine virtuelle Windows avec `qemu-img create -f qcow2 /vservers/debian.qcow2 10G`

2. Comparez les résultats des commandes : `ls -lh /vservers/debian.qcow2`. Est-ce surprenant ?

3. Récupérez l'image ISO.

```
> mkdir /vservers/images
```

```
> cp /media/.../debian.iso /vservers/images
```

4. Lancez l'installation de debian9 avec

```
> kvm -m 1024 -cdrom /vservers/images/debian.iso -boot d /vservers/debian.qcow2
```

Ou

```
> qemu-system-x86_64 -m 2048 -cdrom /vservers/images/debian.iso -boot d /vservers/debian.qcow2
```

5. Une fois l'installation terminée, arrêtez debian

Lancement de Debian

Nous allons lancer debian et tester le réseau auquel il a accès.

1. Lancez debian9 avec `kvm -m 1024 /vservers/debian.qcow2` ou avec la commande `qemu-system-x86_64 -m 1024 /vservers/debian.qcow2`

2. Vérifiez que vous disposez d'un réseau (avec les commandes `ipconfig /all`, `ping`, . . .)

3. Que donne la commande `host www.google.fr` ? Qu'est-ce que cela implique ?

4 Testez `ping 10.0.2.15`. Qui est 10.0.2.15 ?

5. Est ce que vous avez accès au web ?.

6. Par quel procédé technique debian a-t-il accès au Web ?

7. Téléchargez et installez un package avec `apt-get`.

8. Arrivez-vous à ping-er debian depuis la machine hôte ?

Mise en réseau - mode pont

Nous allons maintenant intégrer les machines virtuelles windows et linux à votre réseau local

On va prendre pour l'exemple la machine virtuelle linux mais c'est aussi les mêmes démarches pour windows

Commencez par installer les paquetages `uml-utilities` et `dnsmasq-base`. (sous votre machine Hôte)

Mode pont (bridge) :

1. Éteignez vos machines virtuelles.

2. Sur la machine hôte , nous allons créer un pont qui va contenir à la fois l'interface (physique) `eth0` et l'interface virtuelle de linux (attention, le réseau de votre machine va être coupé lors des commandes suivantes)

NB: Attention l'interface physique peut avoir un autre nom que `eth0` par exemple sur Ubuntu j'ai une interface `ens33`

```
> sudo ifconfig eth0 0.0.0.0
> sudo brctl addbr pont
> sudo brctl addif pont eth0
> sudo ifconfig pont up
> sudo dhclient pont
> sudo tunctl -b -u <user>
```

tunctl vous affiche alors un nom d'interface (du style tap0)

qu'il faut utiliser dans les commandes ci-dessous

```
> sudo brctl addif pont tap0
```

```
> sudo ifconfig tap0 up
```

3. Expliquez chacune des commandes précédentes.

4. Faites un dessin récapitulant la nouvelle configuration du réseau.

Vous pouvez maintenant lancer et tester le réseau.

Test du réseau :

1. Lancez linux avec

```
> qemu-system-x86_64 -m 1024 /vservers/debian.qcow2 -net nic,macaddr=11:22:33:44:55:xx -
net tap,ifname=tap0,script=no
```

(L'adresse MAC à choisir au hasard et qui fera 6 octets, pensez à prendre la valeur XX différente à celle de vos collègues).

2. Vérifiez la configuration du réseau. Est ce que la VM linux a-t-elle récupéré une adresse IP ?

Vérifiez aussi si c'est dans la même plage d'adresse que votre réseau local.

3. Pouvez-vous maintenant faire un ping sur la VM linux depuis la machine hôte ?

4. Essayez de ping er www.google.com depuis votre VM linux .

5. Faites un apt-get update && apt-get install apache2 sur la VM .

Remise en état du réseau :

1. quelles commandes faut-il taper pour remettre votre réseau en état ?

1.7.

réseau – mode NAT

Dans cette partie, vous devez vérifier que vous avez récupéré une configuration normale et que les machines virtuelles sont bien éteintes. Utilisez pour cela les commandes `brctl show`, `ifconfig`, `route -n`, . . .etc.

Nous allons créer un réseau local à la machine dans lequel nous allons mettre les machines virtuelles windows ou linux.

Préparation du réseau

1. Les commandes qui suivent sont à faire en localuser dans la machine

Hôte.

2. Créez un pont dans un nouveau réseau local destiné pour les VM, dans une plage d'adresses que vous choisirez et qui sera différent de la plage d'adresses de votre réseau local, par exemple 192.168.17.0/24

```
> sudo brctl addbr virbr0
```

```
> sudo ifconfig virbr0 up
```

```
> sudo ifconfig virbr0 192.168.17.1 netmask 255.255.255.0
```

3. Lancez `sudo brctl show` et `sudo route -n` .

4. Lancez un serveur DHCP minimal qui fournira des IP à vos machines. Ce serveur ne fonctionne que sur notre réseau local (la commande fait une seule ligne) :

```
> sudo dnsmasq --strict-order --bind-interfaces --conf-file=  
--listen-address 192.168.17.1 --except-interface lo  
--dhcp-range 192.168.17.2,192.168.17.254 --dhcp-lease-  
max=253
```

5. Réservez deux interfaces de type tap, en vous servant de `sudo tunctl -b -u <localuser>`.

L' interface est nommée tap0.

6. Ajoutez cette interface au pont et activez la

```
> sudo brctl addif virbr0 tap0
```

```
> sudo ifconfig tap0 up
```

7. Lancez `sudo brctl show` et `sudo ifconfig` pour contrôler ce qui se passe.

1. Lancement de la VM :

```
>qemu-system-x86_64 -m 1024 /vservers/debian.qcow2 -net  
nic,macaddr=11:22:33:44:55:66 -net tap,ifname=tap0,script=no
```

2. Vérifiez que dnsmasq a bien fourni une IP à v o t r e V M en comparant l'IP donnée par votre

VM et celle dans `/var/log/daemon.log`.

3. Avez-vous accès à la VM depuis la machine hôte ?
4. Avez-vous accès au Web depuis la VM ? Pouvez-vous envoyer un ping sur 192.168.17.1 (i.e. la passerelle qui est la machine hôte) ?
5. Avez-vous accès à internet ? pourquoi ?

Nous allons maintenant autoriser la VM à accéder à l'interface réseau de la machine hôte en utilisant la redirection de ports.

Redirection des ports :

1. Autorisez la redirection avec :

```
> sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
```

2. Activez la translation d'adresses avec :

```
> sudo iptables -t nat -A POSTROUTING -s 192.168.17.0/24  
! -d 192.168.17.0/24 -j MASQUERADE
```