

TD - ETUDE D'UN RÉSEAU EXISTANT

1 Présentation de QEMU

Dans l'UE d'administration des réseaux, nous allons utiliser des machines virtuelles (QEMU) connectées par un réseau virtuel de type `vde`. L'intérêt de l'approche est qu'il suffit d'avoir accès à une seule machine "physique" pour pouvoir faire des manipulations plus ou moins complexes sur un réseau en ayant les privilèges du super-utilisateur (root) sur toutes les machines composant le réseau. Tout ce qui est présenté dans cette partie est fourni à titre informatif. Seul le tout dernier paragraphe mentionne la procédure à suivre pour lancer la plate-forme pour notre TP.

L'environnement QEMU se présente sous la forme d'un exécutable qui prend en argument notamment une partition racine pour "booter" (le noyau du système d'exploitation pouvant soit être inclus dans l'image ou externalisé). La partition racine se présente sous la forme d'une image d'un système standard (contenant tous les outils nécessaires au bon fonctionnement de la machine). Dans notre cas l'image est contenue dans le fichier `/net/stockage/aguermou/qemu/images/AR/2026/debian.img`.

Pour lancer une session "simple" avec une seule machine, il suffit d'appeler l'exécutable QEMU est en lui passant l'image en argument (le tout avec la bonne syntaxe). Étant donné que nos machines physiques sont de type `x86_64` l'exécutable QEMU `qemu-system-x86_64`. Notez qu'il est nécessaire de spécifier la taille maximale de la mémoire qu'on autorise pour cette instance de QEMU (500 Mo dans notre cas). Dans notre cas, ceci donnerait :

```
qemu-system-x86_64 -enable-kvm -hda <nom_image> -m 500M
```

Ceci risque de ne pas bien fonctionner vu que vous n'avez pas les droits UNIX de modifier l'image. Pour résoudre ce genre de problèmes (entre autres), les concepteurs de QEMU ont donné la possibilité aux utilisateurs d'utiliser des fichiers *Copy-On-Write* dont le type est `qcow2`. Ces fichiers peuvent être créés par QEMU pour y stocker les modifications que nous avons apportées à l'image originale (le fichier `qcow2` ne contiendra donc que les différences avec l'image originale). Bien entendu, dans le cas où plusieurs instances de QEMU seraient amenées à s'exécuter simultanément sur une même machine physique, elles ne doivent pas partager le même fichier `qcow2` (ceci est fait pour garantir la consistance du contenu du fichier). Ainsi chaque machine virtuelle QEMU aura son propre fichier `qcow2` spécifié par l'utilisateur. Dans notre cas, si nous voulions utiliser un fichier `qcow2` pour stocker les modifications apportées à notre image, nous devrions tout d'abord le créer en spécifiant l'image originale (en utilisant l'option `-b`) :

```
qemu-img create -b <nom_image> -f qcow2 fichier.qcow2
```

puis en lançant QEMU en lui fournissant comme image le fichier `qcow2` généré :

```
qemu-system-x86_64 -enable-kvm -hda fichier.qcow2 -m 500M
```

Si nous observons maintenant la taille du fichier à l'aide de la commande `ls fichier.qcow2` nous voyons qu'il est de très petite taille en comparaison de l'image originale.

À côté des fonctionnalités décrites ci-dessus, QEMU permet de mettre en relation plusieurs machines virtuelles à l'aide de *switches* virtuels. Ces derniers font partie d'un ensemble d'outils de gestion d'une machine QEMU et seront de type `vde` dans notre contexte. Dans nos TPs, nous utiliserons donc l'outil `vde_switch` outil pour faire communiquer les machines virtuelles entre elles. La création d'un switch `vde` se fait à l'aide de la commande suivante :

```
vde_switch -s nom_switch
```

Il est important de signaler que la création du switch doit être faite avant le lancement des machines qui doivent y être connectées. Pour être reliée au switch, la ligne de commande de lancement d'une instance QEMU doit être enrichie (désolé ça se complique ;-)). Dans notre cas, ceci correspondrait à :

```
qemu-system-x86_64 -enable-kvm -hda fichier.qcow2 -m 500M \
-netdev vde,sock=nom_switch,id=vde0 -device e1000,netdev=vde0,mac=a2:00:00:00:00:01
```

Dans cet exemple, nous disons que la première interface de la machine (dont l'adresse MAC est a2:00:00:00:00:01) est reliée au switch. Notez qu'on peut spécifier le type de l'interface réseau (ici nous choisissons une interface de type Intel e1000).

La topologie réseau correspondante peut être obtenue en lançant le script de démarrage `/net/stockage/aguermou/AR/TP/1/qemunet.sh` en lui fournissant la description de la topologie réseau à l'aide de l'option `-t` ainsi que l'archive contenant la configuration initiale des machines à l'aide de l'option `-a`. Ceci revient à lancer les commandes suivantes:

- au cremi:

```
# cd /net/stockage/aguermou/AR/TP/1/; ./qemunet.sh -x -t topology -a archive_tp1.tgz
```

- à distance:

```
# cd /net/stockage/aguermou/AR/TP/1/; ./qemunet.sh -d tmux -b -t topology -a archive_tp1.tgz
# tmux a
```

Vous pouvez obtenir la liste des options du script en l'exécutant avec l'option `-h`.

Les éditeurs disponibles sur les machines virtuelles sont `vim`, `nano`, et `jmacs/jed` (les deux derniers étant des clones légers d'`emacs`).

2 Configuration des interfaces réseaux

2.1 Commande `ifconfig` / `ip`

1. Connectez-vous en tant que `root` sur `immortal`.
2. Etudiez la commande `ifconfig`. Donnez la liste des interfaces réseaux.
3. Configurez l'interface `eth0` de sorte qu'`immortal` possède l'adresse 172.16.0.1 sur le réseau 172.16.0.0/24.
4. Quel est le type et la classe de cette adresse ?
5. Quelle est l'adresse de broadcast du réseau ?
6. Configurez de manière analogue les 3 autres machines.
7. Vérifiez vos configurations à l'aide de la commande `ping`.
8. Essayez un ping avec l'adresse de broadcast du réseau. Que ce passe-t-il ?
9. Modifier le fichier pour `/proc/sys/net/ipv4/icmp_echo_ignore_broadcasts` pour que les machines répondent à ce ping. Vérifiez.
10. Pour quitter proprement cet environnement QEMU, vous devez lancer la commande `poweroff` sur chacune des machines. Faites-le.

2.2 Fichier `/etc/network/interfaces`

1. Relancer le script `qemunet.sh`. Vérifier que l'interface `eth0` des différentes machines n'est plus correctement configurée.
2. Le fichier `/etc/network/interfaces` permet de configurer de manière permanente les différents interfaces réseaux d'une machine. Remplissez correctement ce fichier pour les différentes machines. Pour cela vous pouvez consulter le man : `man interfaces`.
3. Ce fichier est interprété au démarrage de la machine, ou lorsque vous appelez le script `/etc/init.d/networking restart`. Utilisez la deuxième méthode pour interpréter ce fichier et vérifiez qu'il a bien été pris en compte.
4. Rebootez avec la commande `reboot`
5. Quitter proprement. Normalement, les modifications que vous avez apportées à la configuration de chaque machine sont stockées dans les différents fichiers `.qcow2`. Ils sont stockés dans le dossier `$HOME/qemunet-session`. Pour sauvegarder votre travail, vous pouvez construire une archive contenant l'intégralité du dossier.

```
cd ${HOME}/qemunet-session ; tar cvzf ${HOME}/mysession.tgz * ; cd -
```

N'oubliez pas ensuite de déplacer l'archive quelque part sur votre répertoire d'accueil. Enfin, pour relancer les machines en utilisant ce que vous avez sauvegardé il suffit d'exécuter la commande suivante.

```
cd /net/stockage/aguermou/AR/TP/1/; ./qemunet.sh -x -s ${HOME}/mysession.tgz
```