TP - Virtualisation

Date: 26/11/2018

Présentation du TP

Ce TP a pour objectif de vous familiariser avec la manipulation de machines virtuelles sous Linux avec Qemu/KVM et libvirt. Vous allez prendre en main ces outils et vous confronter aux problématiques de déploiement et gestion d'images ainsi que de configuration réseau associées.

Complétez le google form suivant avec vos réponses:
https://goo.gl/forms/bAKFgRTd77lHXzmu2

Mise en place de l'environnement

Le TP est à réaliser sur le cluster HPC de l'ENSIIE.

Commencez par définir quelques variables d'environnement. Remplacez <login> par votre login sur le cluster.

```
export      SCRATCHDIR=/scratch/<login>
export      TPHOME=/scratch/tp/virt
```

Créez un volume de données persistent partagé entre vos VMs

```
mkdir       $SCRATCHDIR/tp-virt/
truncate    $SCRATCHDIR/tp-virt/tpdata.img --size 20G
```

```
mkfs.ext4  $SCRATCHDIR/tp-virt/tpdata.img
```

Copier le contenu de `$TPHOME/templates.yaml` dans `~/.pcocc/templates.yaml`

Copier et adapter le contenu de `$TPHOME/inituser.ci` dans `$SCRATCHDIR/tp-virt/inituser.ci` (remplacez les TODO par vos login et votre clé publique SSH).

Avec la commande `pcocc batch -t 6:00:00 -c 4 tphost0,tphost1` allouez-vous deux VMs qui représenteront deux hôtes virtuels. Ainsi, dans la suite du TP, lorsque nous parlons du premier hôte nous faisons référence à la VM à laquelle vous vous connectez depuis le cluster HPC de l'ENSIIE en tapant pcocc ssh vm0.

Vérifiez que votre deuxième hôte a bien monté les données partagées en NFS par le premier hôte dans /data. Si ce n'est pas le cas faites le montage avec `mount /data`. Les données dans /data sont persistantes si vous relancez votre cluster pcocc. Stockez vos images de VM et vos scripts dans cet espace partagé. Nous vous conseillons de travailler en root pour la suite du TP (`sudo -i`).

Lancement d'une première machine virtuelle

- A l'aide de l'outil `qemu-img`, créez une image qcow2 de 20GB **(Q1.1)**
- A l'aide de la commande `qemu-system-x86_64`, lancez une machine virtuelle utilisant cette image. La machine virtuelle devra avoir les caractéristiques suivantes:
    a. Machine basée sur le modèle q35
    b. Accélération matérielle à travers l'API kvm
    c. 2 CPUs virtuels, et 4GB de mémoire par VM
    d. Carte réseau Ethernet paravirtualisée virtio-net, avec une adresse MAC que vous définirez. Le réseau Ethernet sera émulé en espace utilisateur.
    e. Périphérique bloc paravirtualisé virtio-blk dont le stockage sera émulé avec l'image qcow2 que vous avez crée
    f. Périphérique console isa-serial qui sera émulé sur la sortie standard de votre terminal
    g. Moniteur qemu multiplexé sur la même sortie standard

- Qemu supporte différentes formes d'arguments en ligne de commande pour définir les périphériques exposés à une VM. Pour cet exercice, utilisez l'option `-nodefaults` pour bien maîtriser la configuration de votre VM et employez les formes d'arguments qui permettent de spécifier séparément le frontend et le backend de chaque périphérique virtuel **(Q1.2)**. Détailler le rôle de chaque argument utilisé **(Q1.3).**

Installation d'une image via kickstart

- Faites démarrer votre VM avec un script ipxe. Votre premier hôte dispose pour cela d'un serveur http exportant les fichiers nécessaires depuis **/var/www/html/deploy**. Vous aurez besoin de l'option `bootfile` du backend réseau user **(Q2.1)**.
- Complétez les fichiers ipxe et kickstart modèles fournis dans ce répertoire **(Q2.2)**. Vous devez vous assurer qu'il vous sera possible de vous connecter à la VM déployée en root avec une clé SSH que vous définirez, et que la machine pourra se connecter à elle même en SSH avec une clé privée déployée dans le kickstart. Définissez aussi un mot de passe root pour pouvoir vous connecter depuis la console en cas de problème.
- Connectez-vous à votre VM via votre console série avec un mot de passe. Depuis votre VM, vérifiez que vous arrivez à contacter le deuxième hôte en utilisant la commande `nc` (créez un serveur nc sur le deuxième hôte avec `nc -v -l <port>` et connectez y vous avec `nc -v <ip> <port>`). Avec la commande `tcpdump`, indiquez les adresses MAC, IP et ports (source et destination) des paquets émis par la machine virtuelle. A qui correspondent ils ? **(Q2.3)** Idem pour les paquets reçus par le deuxième hôte. **(Q2.4)**
- Comment se connecter à votre machine virtuelle via SSH depuis votre machine hôte ? Regardez l'option hostfwd du backend réseau user. Donnez les arguments Qemu et SSH utilisés **(Q2.5, Q2.6)**

Lancement d'un petit cluster de VMs

- Avec la commande brctl, ajoutez un périphérique bridge à chacune de vos machines hôte et connectez les au réseau physique accessible via eth1 **(Q3.1)**.
- Choisissez une plage réseau IP pour ce réseau Ethernet et assignez une IP à chacun de vos hôtes **(Q3.2)**. Vérifiez que vos deux hôtes arrivent à se joindre sur ce réseau.
- Avec `qemu-img`, créez trois nouvelles images qcow2 de 20GB utilisant votre première image comme image de référence **(Q3.3)**. Expliquer pourquoi il est judicieux de positionner l'image de référence en lecture seule **(Q3.4)**.

- L'objectif est de lancer trois VMs à partir de ces images, deux sur le premier hôte et une sur le deuxième hôte. Chacune de ces VMs doit disposer d'une seconde interface Ethernet virtuelle connectée au réseau Ethernet physique accessible via l'interface eth1 des hôtes. Utilisez pour cela le backend réseau tap. Faites un script permettant de lancer une machine virtuelle à partir des arguments que vous jugerez nécessaires et lancez vos 3 VMs **(Q3.5, Q3.6)**.
- Connectez-vous y et assignez leur une IP dans la plage réseau que vous avez définie **(Q3.7)**. Vérifiez que vous pouvez vous connecter en SSH à vos VMs en passant par ce réseau, et que vos VMs peuvent se connecter entre elles via SSH.
- Précisez le chemin suivi par un paquet réseau envoyé entre deux VM sur le même hôte, et deux VMs sur deux hôtes différents **(Q3.8)**. Quelles adresses MAC, IP et port (source et destination) sont indiqués dans les paquets émis par la VM source et lorsqu'il traversent les différents composants réseaux physiques et virtuels **(Q3.9)**. Expliquer pourquoi l'interface eth1 de vos hôtes est configurée en mode promiscuous lorsqu'elle est connectée au bridge. **(Q3.10)**

Gestion des images

- Notez la taille que vos fichiers d'image de VMs occupent sur disque. Expliquer pourquoi les tailles indiquées par les commandes `du` et `ls` peuvent être différentes **(Q4.1).**
- Écrivez un fichier de quelques GBs dans une de vos VM (par exemple avec `dd if=/dev/zero of=filename bs=1M count=2000`) et supprimez le. Comment évolue l'espace disque utilisé par le système de fichiers dans la VM ainsi que la taille du fichier d'image virtuel ? **(Q4.2)**
- Expliquer pourquoi et donner une solution en lisant la page de man du système de fichiers utilisé dans votre VM **(Q4.3)**. Expliquer pourquoi cela nécessite de remplacer le modèle de périphérique virtuel bloc par un périphérique virtio-scsi **(Q4.4).** Relevez à nouveau les tailles de votre image **(Q4.5)**.
- Effacez les fichiers kernel (/boot/vmlinux*) d'une de vos VMs. Après un reboot, la VM ne démarre plus. Utilisez les outils du paquet libguestfs pour réparer votre image de VM (regardez les commandes guestfish, et virt-*) **(Q4.6).**

Configuration d'images cloud

- On souhaite recréer ce cluster de VMs mais en s'appuyant cette fois sur cloud-init et des images cloud plutôt que sur une image déployée via kickstart. On prendra pour image de référence l'image cloud `$TPHOME/cloud-centos7.5.qcow2` que vous copierez dans l'espace `/data` sur vos hôtes. Modifiez votre script de lancement de VMs pour qu'elles utilisent un disque dur éphémère basé sur cette image (créez une image temporaire à chaque lancement de VM et supprimez la lorsque la VM est terminée).
- Une VM cloud a besoin de configuration pour démarrer correctement. Ecrivez un fichier user-data déployant une configuration s'approchant de celle de votre kickstart **(Q5.1)**.
- Configurez-vos VMs à la volée grâce à la source de données NoCloud acceptée par cloud-init qui permet de passer une configuration via un cdrom virtuel (https://cloudinit.readthedocs.io/en/latest/topics/datasources/nocloud.html). Modifiez votre script pour générer un fichier metadata et un cdrom à la volée et lancez vos 3 VMs **(Q5.2, Q5.3)**.

Routage

- On souhaite maintenant se passer du réseau Ethernet émulé en espace utilisateur. Relancez vos VM avec uniquement l'interface Ethernet virtuelle connectée au réseau physique eth1. Plusieurs problèmes se posent:
    - Au démarrage, une VM essaie de récupérer une IP en dhcp, mais il n'y a pas de serveur dhcp sur ce réseau physique . Sur votre premier hôte ajoutez un serveur dnsmasq qui ecoute sur ce réseau physique et distribue des adresses IPs à vos VMs (consulter le man de dnsmasq) **(Q6.1)**.
    - Vos VMs n'ont plus de routeur par défaut et n'ont donc plus d'accès hors de leur sous-reseau IP. Nous allons donc configurer notre premier hôte comme passerelle NAT. Autorisez votre premier hôte à router les paquets qui lui sont adressés (voir: https://www.kernel.org/doc/Documentation/networking/ip-sysctl.txt). Avec la commande iptables, ajoutez une règle dans la chaîne POSTROUTING de la table nat (appliquée aux paquets que votre hôte va router) réalisant une traduction d'adresse source entre les IP appartenant du sous-réseau choisi pour vos VM et l'IP publique de votre premier hôte **(Q6.2)**

- Depuis une VM exécutée sur le deuxième hôte, connectez vous à un noeud de login du cluster de l'ENSIIE via nc. Donnez les adresses MAC, IP et ports, (source et destination) indiqués dans les paquets émis par la machine virtuelle et dans ceux émis par le premier hôte **(Q6.3,6.4)**.
- Quelle IP source voit votre serveur nc sur le noeud de login du cluster et pourquoi à votre avis ? **(Q6.5)**

Utilisation de libvirt

- Avec la commande `virt-install` importez deux images que vous aviez déployé avec kickstart dans libvirt sur le premier hyperviseur. Utilisez un réseau connecté au bridge que vous avez défini dans les parties précédentes et utilisez le mode de cache ***none*** pour le backend de votre disque. **(Q7.1)**
- Étudiez la commande virsh permettant d'interagir avec libvirt. Regardez comment lister vos VMs en cours d'exécution, vous connecter à leur console et afficher des statistiques. Affichez la représentation XML de votre VM.
- Faites une migration à chaud de votre seconde VM vers votre second hôte et vérifiez que cela est bien transparent pour un utilisateur de votre VM. **(Q7.2)**
- Proposez une méthode pour mesurer la durée d'indisponibilité lors de la migration et indiquez le résultat dans diverses configurations. **(Q7.3)**