Année Universitaire 2015-2016
3ième année IMAC

Mardi 5 janvier 2016

Cloud computing
Travaux Pratiques

Objectif

Dans un premier temps, on utilisera libvirt : une librairie d'accès aux principaux hyperviseurs du marché ; l'API libvirt est écrite en C et est sous licence libre (http://libvirt.org/). Nous allons utiliser le programme `virsh`, une interface en ligne de commande pour la librairie libvirt pour manipuler les machines virtuelles : création, démarrage, arrêt, destruction, modification des ressources, snapshot, recopie et déplacement.

Dans un second temps, nous installerons dans une machine virtuelle, un service de stockage de fichiers en ligne. On profitera de ces dispositifs pour analyser les traces sur le réseau, ce qui nous conduira à augmenter le niveau de sécurité du service. Enfin nous ferons une analyse de traces de serveur.

En début de travaux pratiques, un étudiant utilise un poste de travail et pour la partie de déplacement des machines virtuelles, il faut travailler avec deux postes et donc par groupe de deux étudiants.

Installations préalables

Se logguer avec l'utilisateur `tpreseau`, mot de passe `tpreseau`, ouvrir un terminal de commande.
Vérifier les possibilités de virtualisation du CPU, la commande

```
egrep '(vmx|svm)' --color=always /proc/cpuinfo
```

doit afficher les caractères `vmx` ou `svm` en couleur.
Vérifier la configuration BIOS, la commande

```
dmesg | grep -i kvm
```

ne doit rien afficher.
Pour avoir les droits root avec transfert des droits sur le serveur X, exécuter la commande,

```
sux
```

le mot de passe est `tpreseau.`
Les fichiers nécessaires au TP sont dans le répertoire /mnt/nfs, déplacer les scripts et fichiers de configuration dans /root

```
mv /mnt/nfs/jessie.xml /mnt/nfs/*.sh /mnt/nfs/*.php /root
```

Faire tourner le script

```
./generic-init.sh
```

Obtenir de l'aide sur l'interface `virsh` :

```
virsh help|less
```

Obtenir de l'aide sur la commande `start` :

```
virsh help start
```

Le cycle de vie des domaines

Éditer le fichier `jessie.xml` pour lui attribuer une adresse mac en remplaçant les xx par votre numéro puis démarrer le domaine jessie qui n'est pas encore enregistré dans la liste des domaines gérés par libvirt

```
virsh create jessie.xml
```

Accéder à la console du domaine jessie

```
virt-viewer jessie &
```

Vérifier le démarrage

```
virsh list --all
```

Arrêter le domaine

```
virsh shutdown jessie
```

Vérifier l'arrêt

```
virsh list --all
```

Enregistrer le domaine jessie dans la liste des domaines gérés par libvirt

```
virsh define jessie.xml
```

Vérifier cet enregistrement

```
virsh list --all
```

Supprimer cet enregistrement du domaine jessie

```
virsh undefine jessie
```

Vérifier la suppression

```
virsh list --all
```

Définir à nouveau le domaine jessie

```
virsh define jessie.xml
```

Démarrer le domaine jessie qui est défini (noter la différence avec la commande « virsh create jessie.xml » qui permet de démarrer un domaine qui n'a pas été défini)

```
virsh start jessie
```

Vérifier le démarrage

```
virsh list
```

Vérifications

Accéder à la console de la machine invité

```
virt-viewer jessie &
```

Se logguer en tant `root`, mot de passe `tpreseau` et lancer la commande

```
ip a
```

pour récupérer l'adresse IP du domaine invité.

Utiliser ssh

Afin de simplifier les commandes et en même temps de montrer l'utilisation des commandes ssh et pipe, nous allons permettre à l'utilisateur root de la machine hôte de se connecter en ssh sur le compte root de la machine invité sans avoir à rentrer de mot de passe.
Il faut créer un couple de clé privé, clé public sur la machine hôte :

```
ssh-keygen
```

- le choix proposé d'emplacement de la clé peut être validé, appuyer sur entrée pour le valider
- la passphrase peut être vide, appuyer sur entrée
- pour confirmer, appuyer à nouveau sur entrée

Il faut envoyer la clé public de root de la machine hôte dans le compte root de la machine invité, pour des raisons de sécurité, on ne peut pas se connecter directement avec le compte root, il faut donc passer par le compte tpreseau :

1) copier la clé public de root de la machine hôte dans le compte tpreseau de la machine invité :

```
cat /root/.ssh/id_rsa.pub | ssh tpreseau@ip_inv "cat >> authorized_keys"
```

2) installer la clé dans le compte root de la machine invté

```
ssh tpreseau@ip_inv
su
mv /home/tpreseau/authorized_keys /root/.ssh/
```

```
chown root:root /root/.ssh/authorized_keys
chmod 600 /root/.ssh/authorized_keys
```

Depuis le compte root de la machine hôte, réaliser une première connexion ssh
```
ssh ip_inv
```
afin de stocker une empreinte (le fingerprint) de la machine invité sur votre machine. Cette empreinte permet de se protéger d'une usurpation d'identité de la machine cible, ip_inv dans ce cas. Il s'agit d'une ligne stokée dans le fichier /root/.ssh/known_hosts
Vous pouvez sortir avec la commande
```
exit
```

Vous pouvez maintenant vérifier que la commande
```
ssh ip_inv
```
ne demande pas de mot de passe. Puis sortir du terminal invité avec
```
exit
```

Installer le script heure.php qui donne l'heure sur la machine invité, dans le répertoire
```
/var/www/html/
```
de la machine invité

Et dans un navigateur de la machine hôte, aller à l'adresse
http://ip_inv/heure.php

Sauvegarde à chaud

Pour réaliser une sauvegarde à chaud des domaines, on exécute les trois opérations suivantes :
1) figer le domaine
2) prendre un snapshot (une photo) de la partition support du domaine
3) reprendre l'exécution normale du domaine.
Nous n'avons pas la possibilité ici de montrer le snapshot de la partition, nous ne montrons donc que les étapes 1) et 3), pendant ces deux opérations, conserver un regard sur http://ip_inv/heure.php

Suspendre l'exécution du domaine
```
virsh suspend jessie
```

Reprendre l'exécution du domaine
```
virsh resume jessie
```

Modifier les ressources

Dans le domaine jessie, exécuter la commande top afin de visualiser les ressources du domaine
```
top
```

Puis, sur l'hyperviseur, augmenter la mémoire de la façon suivante
```
virsh setmem jessie 1048576
```
Vous devez constater une augmentation immédiate de la ressource Mem, il est aussi possible de diminuer la mémoire, à exécuter avec la plus grande précaution, car si un processus utilise la partie de la mémoire qui disparaît du système, ce processus risque fort de disparaître !

Augmenter le nombre de CPU :
Arrêter la machine jessie et éditer son fichier de configuration :
```
virsh edit jessie
```
Modifier la ligne
```
  <vcpu placement='static'>1</vcpu>
```
en

<vcpu placement='static'>2</vcpu>

Redémarrer le domaine pour visualiser l'apparition d'un second CPU, exécuter la commande top, puis appuyer sur la touche 1.

Pour ne pas surcharger le système inutilement, refaire l'opération avec <vcpu placement='static'>1</vcpu>

Migration de domaine

Il faut maintenant travailler avec deux postes et donc à deux étudiants, soient les postes A et B. L'objectif de cet exercice est de montrer que l'image (le disque virtuel) d'un domaine peut résider sur un système de fichiers et qu'il peut être exécuté sur une autre machine hôte ; on va montrer que la migration du domaine peut se faire sans arrêt du domaine. Cet exercice illustre les propos d'Eben Moglen : « Cloud means servers have gained freedom. Freedom to move. Freedom to dance; to combine and to separate, re-aggregate, and do all sorts of tricks. » lors d'une conférence donnée à l'Internet Society en 2010. Ici, l'image du domaine réside sur la machine A. Au début de l'exercice, le domaine est exécuté sur la machine A et nous montrons que tout en résidant sur la machine A son exécution peut se déplacer sur la machine B.

Situation initiale :

Sur la machine A

Le domaine jessie est défini est il est en cours d'exécution, vérifier avec

```
virsh list
```

Il faut installer un serveur de fichiers pour permettre à la machine B d'accéder à l'image du domaine. Avec le script suivant, nous installons le serveur NFS (Network File System)

```
./serveur-nfs.sh
```

Sur la machine B

Le domaine jessie n'est pas défini, exécuter

```
virsh shutdown jessie
virsh undefine jessie
```

Pour que le poste B accède au serveur de fichiers de la machine A, il faut éditer le script `client-nfs.sh` et indiquer l'adresse IP de la machine A, puis faire tourner le script

```
./client-nfs.sh
```

La commande

```
mount
```

vous permettra de vérifier le montage de la partition

Pour la suite, vous pouvez copier la clé ssh publique de l'utilisateur root de la machine A dans le compte root de l'utilisateur root de la machine B comme vue précédemment.

Maintenant les deux machines sont prêtes pour la migration du domaine, pendant l'opération observez bien la continuité d'exécution du domaine http://ip_inv/heure.php
sur la machine A, exécuter la commande

```
virsh migrate --live jessie qemu+ssh://ip_de_B/system
```

Sur les machines A et B, vérifier la liste des domaines

```
virsh list --all
```

A la fin de cet exercice, pour continuer dans de bonnes conditions sur la machine B, il est nécessaire de démonter le volume NFS avec la commande

```
umount /mnt/nfs
```

Pour vérifier que la commande a été effectuée avec sucée, lancer la commande

```
mount | grep nfs
```

qui ne doit rien renvoyer

Recopie des domaines

Une fonctionnalité très appréciée de ces domaines est de pouvoir se répliquer très facilement. On montre dans ce TP deux modes de recopies.

1) Le clonage d'un domaine, il s'agit d'une recopie de l'image dans l'état, l'outil de création prend soin de ne pas recopier les informations qui doivent être uniques pour un domaine.
2) Le snapshot n'est pas une véritable recopie, il s'agit d'une photo du système à un instant T. Il permet de conserver l'état d'un domaine à un instant T.

Le clonage d'un domaine

Attention, la recopie ne peut pas se faire à chaud

```
virsh shutdown jessie
```

définissez votre adresse mac en remplaçant les xx par votre numéro

```
virt-clone --original=jessie --mac 54:52:00:03:xx:00 --auto-clone
```

Vous venez d'installer une machine complète en un temps record ! Vérifiez que les domaines démarrent bien

```
virsh start jessie
virsh start jessie-clone
```

Pour visualiser les différences des deux domaines, vous pouvez exécuter la séquence de commande suivante :

```
virsh dumpxml jessie > /tmp/jessie.xml
virsh dumpxml jessie-clone > /tmp/jessie-clone.xml
diff /tmp/jessie.xml /tmp/jessie-clone.xml
```

Snapshot d'un domaine

Sur la machine invitée, écrire un fichier

```
echo "avant snapshot" > journal
```

vérifier l'écriture de ce fichier :

```
cat journal
```

Voir la liste des snapshot d'un domaine

```
virsh snapshot-list jessie
```

Créer un snapshot

```
virsh snapshot-create jessie
```

noter le numéro du snapshot crée

vérifier la présence du snapshot

```
virsh snapshot-list jessie
```

effectuer une modification dans le domaine

```
echo "apres snapshot" >> journal
```

vérifier le contenu du fichier :

```
cat journal
```

revenir au point du snapshot

```
virsh snapshot-revert jessie xxx
```

où xxx est le numéro du snapshot que l'on souhaite restaurer

vérifier dans la VM

```
cat journal
```

supprimer le snapshot
```
virsh snapshot-delete jessie xxx
```
où xxx est le numéro du snapshot que l'on souhaite effacer

Vérifier la disparition du snapshot
```
virsh snapshot-list jessie
```

Travailler sur le disque virtuel

Travailler sur un disque dont le système est éteint
```
virsh shutdown jessie
```
Préparer le point de montage
```
mkdir /mnt/test
```
Vérifier que le répertoire est bien vide
```
ls /mnt/test
```
Monter le disque
```
guestmount -d jessie -i /mnt/test
```
Vérifier que le disque est bien monté
```
ls /mnt/test
cat /mnt/test/root/journal
```
Démonter le disque
```
guestunmount /mnt/test
```

Installation de ownCloud

ownCloud est un logiciel libre de service de stockage et de partage de fichiers en ligne qui possède des clients intégrés pour les principaux systèmes d'exploitation windows, mac et linux et pour les mobiles android et ios ; on peut comparer son périmètre fonctionnel à Dropbox. Il existe d'autres logiciels libres sur ce sujet : sparkleshare, seafile, cozycloud et pydio.

Pour gagner du temps, ownCloud est déjà installé sur la machine invité, l'installation a été faite tout simplement en utilisant le gestionnaire de paquet de la distribution Debian de cette façon :
```
root@jessie:~# apt-get install owncloud mysql-server
```
Il reste toutefois à installer une base de données pour contenir les données de l'application, vous pouvez paramétrer le script install_owncloud_database.sh pour continuer l'installation.

Vérifier l'installation avec l'url http://ip_inv/owncloud

créer le compte administrateur, par exemple :
utilisateur = admin , mot de passe = tpreseau20160104

puis renseigner les champs relatifs à la base de données en prenant soin de les faire correspondre aux paramètres du script install_owncloud_database.sh

La création des comptes utilisateurs se fait dans le menu admin (en haut à droite de l'écran) et dans la partie Utilisateurs ; créer un compte toto et un autre titi, sortir de la session admin, rentrer avec la session toto, uploader un document, le partager avec titi, sortir de la session, rentrer avec titi et vérifier que le document uploadé est bien disponible, puis sortir de l'application.

Le site protège son accès par mot de passe sans chiffrement, c'est comme fermer une porte sans la fermer à clé, n'importe qui peut rentrer sans casser la serrure. Pour le montrer, il suffit de se placer sur une machine qui se trouve sur le chemin réseau de la requête (en l'occurrence la machine hôte ou la machine invité) et observer le flux :

avec la commande sur la machine invité :
```
tcpdump -i eth0 port 80 -A | grep -i password=
```
et sur la machine hôte avec la commande :

```
tcpdump -i br0 port 80 -A | grep -i password=
```

si la commande tcpdump n'est pas installée, vous pouvez l'installer avec la commande

```
apt-get install tcpdump
```

À la suite de cette observation, la nécessité de chiffrer le flux réseau devient évidente. Il faut dans un premier temps installer le module SSL du serveur Apache, le paramétrer et installer un certificat. Comme il n'est pas possible de réaliser un certificat chez un tiers de confiance dans le cadre de notre TP, nous nous contenterons donc d'un certificat auto-signé. Dans un second temps, il faut aller dans l'interface d'administration d'ownCloud pour cocher la case « Forcer HTTPS ».

Le script install_ssl-owncloud.sh vous permettra de réaliser cette installation. Comme l'opération de création du certificat auto-signé génère un ensemble de questions et réponses, il est préférable de copier le script install_ssl-owncloud.sh sur la machine invité et d'exécuter le script en local :

depuis la machin hôte :

```
scp install_ssh-owncloud.sh root@ip_inv:
```

depuis la machine invité :

```
sh install_ssh-owncloud.sh
```

À la suite de cette installation, vérifier que le chiffrement est bien opérationnel :

```
tcpdump -i eth0 port 80 or port 443 -A | grep -i password=
```

Analyse des traces

Donner la liste des adresses IP qui ont vu la page truc.html

```
grep truc.html access.log
grep truc.html access.log | cut -d" " -f1
grep truc.html access.log | cut -d" " -f1 | sort
grep truc.html access.log | cut -d" " -f1 | sort | uniq -c
grep truc.html access.log | cut -d" " -f1 | sort | uniq -c | sort -rn
```

utiliser man grep, man cut, man sort, man uniq pour bien comprendre chacune des étapes

Obtenir la liste des adresses IP qui se sont connectées le 8 février entre 7h et 7h15 et le nombre de fois où chacune d'elle s'est connectée. Avec la suite des commandes suivantes, expliquer comment on arrive au résultat.

```
grep "08/Feb/2014:07:0" access.log
grep "08/Feb/2014:07:1[0-5]" access.log
grep "08/Feb/2014:07:0\|08/Feb/2014:07:1[0-5]" access.log
grep "08/Feb/2014:07:0\|08/Feb/2014:07:1[0-5]" access.log | cut -d" " -f1 |sort
|uniq -c
```

Obtenir la liste des adresses ip qui se sont connectées le 9 février entre 7h05 et 7h15 et qui ont vu la page bidule.html et le nombre de fois où ces deux conditions se sont produites.

Donner tous les jours où l'adresse IP 192.168.11.53 s'est connecté. Utiliser la commande man sed

```
grep 192.168.11.53 access.log
grep 192.168.11.53 access.log| sed -e 's/.*\[//'
grep 192.168.11.53 access.log| sed -e 's/.*\[//' -e 's/:.*//'
grep 192.168.11.53 access.log| sed -e 's/.*\[//' -e 's/:.*//'|uniq -c
```