Table des matières

1) Objectifs du TP et organisation....1
2) Pré-requis, recommandations et notation du TP....1
3) Environnement du TP....2
3.1) Installations nécessaires au TP....2
4) Création de machines virtuelles K-Vms....3
4.1) Création de VMs avec virt-manager....3
4.1.1) Accès à virt-manager....3
4.1.2) Installation d'une VM Centos avec virt-manager....3
4.2) Création d'une K-VM Debian avec virt-install....6
5) Découverte de l'architecture KVM....9
5.1) Gestion du réseau....9
5.3) Clonage d'une K-VM et gestion du stockage....10

1) Objectifs du TP et organisation

— Comprendre l'architecture d'une solution de virtualisation.

— Créer une machine virtuelle Kvm à l'aide de virt-manager et de virt-install.

— Utiliser la libvirt au travers du shell virsh.

— Mettre en réseau une machine virtuelle de différentes façons.

— Migrer une machine virtuelle d'un noeud KVM à un autre en utilisant un espace de stockage NFS

2) Pré-requis, recommandations et notation du TP

Les pré-requis sont les suivants :

— Avoir un PC sous Linux.

— La virtualisation consomme de la mémoire. Plus vous en aurez et mieux cela sera.C'est l'enseignant qui décide du nombre de personnes dans un groupe de travail (de 1 à n). Il peut vous être explicitement demandé de faire valider votre

travail au cours du TP par l'enseignant au fur et à mesure de votre avancement pour être noté. Un compte rendu réalisé au fil du TP est obligatoire. Faites impérativement un compte rendu au fur et à mesure avec des copies d'écran et les configurations mises en oeuvre. Tous les travaux sont à déposer sur l'ENT indiqué par l'enseignant. Un travail doit être enregistreravec les noms des personnes dans le nom du fichier, et l'intitulé du fichier doit être clair (par ex :TP_intitulé_du_tp_Etudiant1_Etudiant).Les délais sont parfois et exceptionnellement négociables mais une fois fixés doivent être respectés sous peine d'une note nulle.

3) Environnement du TP

3.1) Installations nécessaires au TP

Nous allons travailler en "Nested Virtualization" avec VMWare Workstation ou autrement dit on va créer des machines virtuelles qui pourront à leurs tours servir d'hyperviseur.Par convention dans ce document les machines virtuelles sont désignées par le sigle VM (VMs au pluriel). Une machine créée avec KVM sera désignée par le sigle K-VM.Vous utiliserez l'OVA fournie par l'enseignant qui contient un serveur NFS :Cette VM contient des images ISO (net -install) et permet de stocker les images des K-VMs afin de réaliser une migration d'une K-VM d'un noeud à un autre.Renseignez dans chaque fichier /etc/hosts des VMs leurs IP respectives.Pour accéder au partage NFS vous devrez passer plus tard au cours du TP (Cf 5.2) ces commandes :

```
mkdir /var/lib/libvirt/images/nfs4mount -t nfs4 -o sec=sys
nfsserveur.example.com:/kvm /var/lib/libvirt/images/nfs4
```

Valider que votre machine supporte KVM :

```
root@ubuntu:~# grep -E '(vmx|svm)' /proc/cpuinfo
root@ubuntu:~# virt-host-validate
  QEMU: Checking for hardware virtualization
: FAIL (Only emulated CPUs are available, performance will be
significantly limited)
  QEMU: Checking if device /dev/vhost-net exists
: PASS
  QEMU: Checking if device /dev/net/tun exists
: PASS
  QEMU: Checking for cgroup 'cpu' controller support
: PASS
  QEMU: Checking for cgroup 'cpuacct' controller support
: PASS
  QEMU: Checking for cgroup 'cpuset' controller support
: PASS
```

```
  QEMU: Checking for cgroup 'memory' controller support
: PASS
  QEMU: Checking for cgroup 'devices' controller support
: PASS
  QEMU: Checking for cgroup 'blkio' controller support
: PASS
...
```

4) Création de machines virtuelles K-Vms

4.1) Création de VMs avec virt-manager

4.1.1) Accès à virt-manager

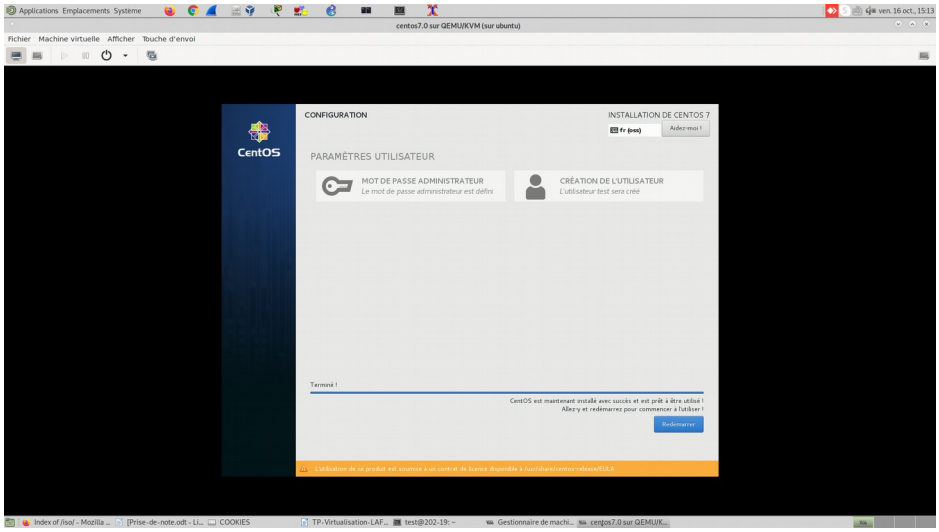
Il faut aussi que le port série de la console soit configuré dans la K-VM :

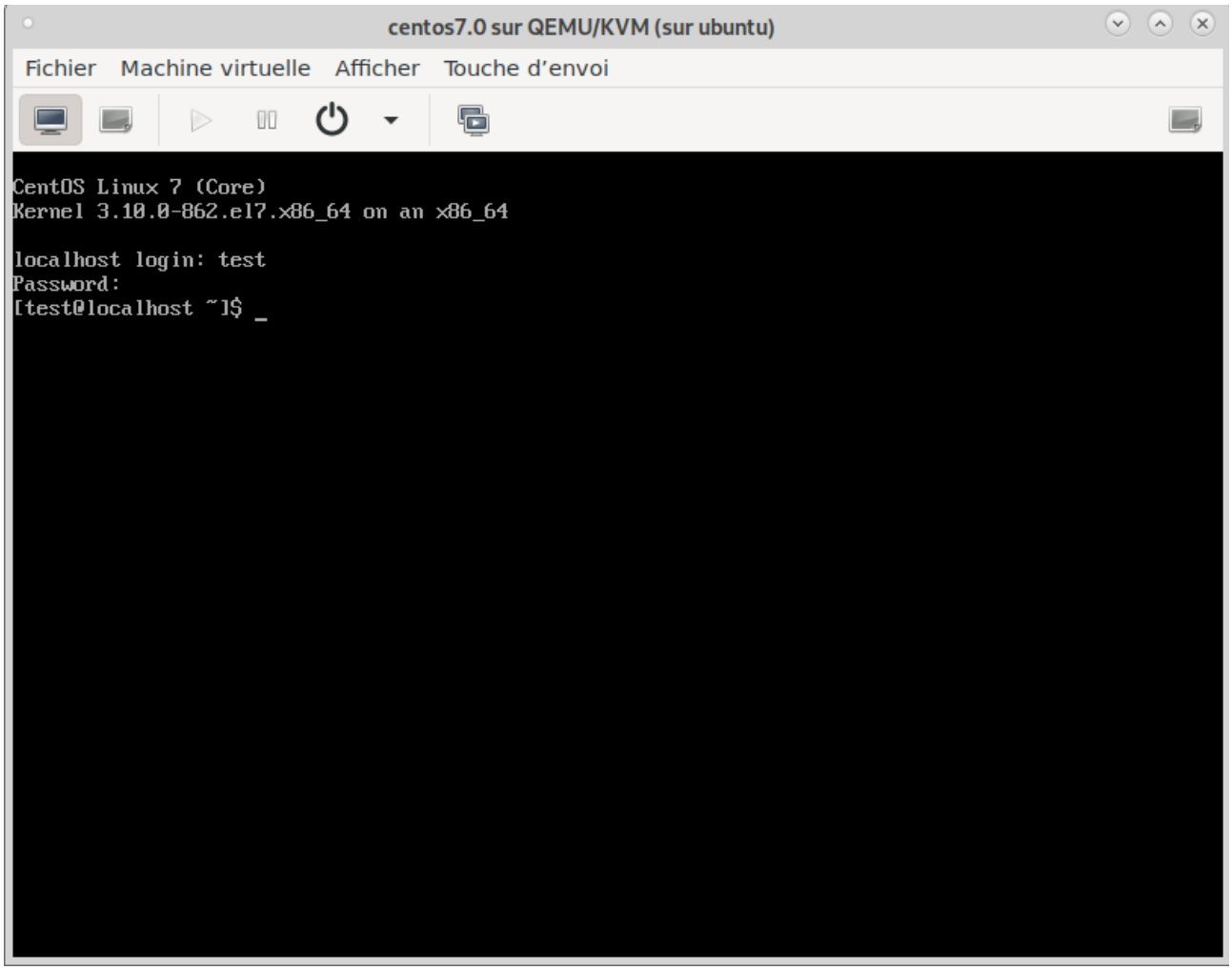
```
systemctl start serial-getty@ttyS0
systemctl enable serial-getty@ttyS0
```

4.1.2) Installation d'une VM Centos avec virt-manager

J'installe une VM CentOS 8 grâce à une image iso récupérée sur le store de l'iut.

Je commence par lancer virt-manager graphiquement dans ma VM et je sélectionne installer une VM via une ISO (je dois scp mon iso qui se trouve sur ma machine physique ou installer via wget l'image ou directement mettre le lien vers l'iso) puis je lui donne une configuration (2 CPU, 5Go de Stockage et 2Go de RAM)





Ma VM a bien été installé et est fonctionnelle

Je me place dans le virsh pour pouvoir avoir les infos de ma K-VM :

```
root@ubuntu:~# virsh
Welcome to virsh, the virtualization interactive terminal.

Type:  'help' for help with commands
       'quit' to quit
```

1) Voici la commande pour récupérer les infos de ma node :

```
virsh # nodeinfo
CPU model:           x86_64
CPU(s):              8
CPU frequency:       2496 MHz
CPU socket(s):       8
Core(s) per socket:  1
Thread(s) per core:  1
```

```
NUMA cell(s):          1
Memory size:           9960324 KiB
```

2) Je liste les domaines actifs de mon K-VM :

```
virsh # list --state-running
 Id   Name         State
----------------------------
 4    centos7.0    running
```

3) Maintenant les domaines actifs et inactifs :

```
 virsh # list --all
 Id   Name         State
----------------------------
-    centos7.0    shut off
```

On ne voit pas de domaines actifs car je n'ai pas d'autres K-VM

4) Je démarre ma K-VM :

```
virsh # list --all
 Id   Name         State
----------------------------
 4    centos7.0    running
```

5) Pour auto exécuter ma K-VM au lancement de ma VM ubuntu je fais la commande suivante :

```
virsh # autostart centos7.0
Domain centos7.0 marked as autostarted
```

6) Je récupère des infos sur ma K-VM avec son domaine :

```
virsh # dominfo centos7.0
Id:             5
Name:           centos7.0
UUID:           fa8af440-9dd9-4437-8a3b-8a71a483c668
OS Type:        hvm
State:          running
CPU(s):         2
CPU time:       25,5s
Max memory:     2097152 KiB
Used memory:    2097152 KiB
Persistent:     yes
Autostart:      enable
Managed save:   no
```

```
Security model: apparmor
Security DOI:   0
Security label: libvirt-fa8af440-9dd9-4437-8a3b-8a71a483c668
(enforcing)
```

7) Je shutdown ma K-VM :

```
virsh # shutdown centos7.0
Domain centos7.0 is being shutdown
```

Maintenant je rdémarre ma K-VM :

```
virsh # start centos7.0
Domain centos7.0 started
```

4.2) Création d'une K-VM Debian avec virt-install

1) Voici la commande pour installer une K-VM debian10 avec virt-install :

```
root@ubuntu:~# virt-install --name debian10 --ram 2048 --disk
path=/var/lib/libvirt/images/debian10.qcow2,size=5 --vcpus 1 --
location
'http://ftp.fr.debian.org/debian/dists/stable/main/installer-
amd64/'
WARNING  Aucun système d'exploitation détecté, la performance de
la machine virtuelle peut en être affectée. Spécifier un système
d'exploitation avec --os-variant pour obtenir des résultats
optimaux.

Début d'installation…
Récupération du fichier linux…
| 5.0 MB  00:00:04
Récupération du fichier initrd.gz…
|  29 MB  00:00:06
Allocation de « debian10.qcow2 »
| 5.0 GB  00:00:00
```

Une fois la commande lancé une K-VM debian s'ouvre et il faut la configurer graphiquement

2) Je récupère des infos sur ma K-VM debian :

```
virsh # dominfo debian10
Id:             8
Name:           debian10
UUID:           a861f2ad-633d-440b-8d7e-c6b3499e021b
OS Type:        hvm
State:          running
CPU(s):         1
CPU time:       31,5s
Max memory:     2097152 KiB
Used memory:    2097152 KiB
Persistent:     yes
Autostart:      disable
Managed save:   no
Security model: apparmor
Security DOI:   0
Security label: libvirt-a861f2ad-633d-440b-8d7e-c6b3499e021b
(enforcing)
```

```
virsh # domilist debian10
error: unknown command: 'domilist'
virsh # schedinfo debian10
Scheduler      : posix
cpu_shares     : 1024
vcpu_period    : 100000
vcpu_quota     : -1
emulator_period: 100000
emulator_quota : -1
global_period  : 100000
global_quota   : -1
iothread_period: 100000
iothread_quota : -1
```

```
virsh # vcpucount debian10
maximum      config         1
maximum      live           1
current      config         1
current      live           1
```

```
virsh # domiflist debian10
 Interface   Type      Source    Model   MAC
-------------------------------------------------------
 -           network   default   e1000   52:54:00:d8:dd:7f
```

3) Pour modifier à froid le nombre de vCPU de ma K-VM je dois l'éteindre puis quitter le shell virsh et lancer la commande suivante :

```
root@ubuntu:~# virsh edit debian10

Select an editor.  To change later, run 'select-editor'.
  1. /usr/bin/vim.basic
  2. /usr/bin/vim.tiny

Choose 1-2 [1]: 1
Domain debian10 XML configuration edited.
```

Je sélectionne l'entrée 1 et ça me renvoi dans un fichier vim où il y a la configuration de ma K-VM. Je modifie le nombre de vCPU dedans directement puis je sauvegarde :

```
<domain type='kvm'>
  <name>debian10</name>
  <uuid>a861f2ad-633d-440b-8d7e-c6b3499e021b</uuid>
  <memory unit='KiB'>2097152</memory>
  <currentMemory unit='KiB'>2097152</currentMemory>
  <vcpu placement='static'>3</vcpu>
  <os>
    <type arch='x86_64' machine='pc-i440fx-focal'>hvm</type>
    <boot dev='hd'/>
  </os>
  <features>
    <acpi/>
    <apic/>
    <vmport state='off'/>
```

Maintenant je retourne dans le shell virsh et je vérifie le nombre de vCPU de ma K-VM :

```
virsh # dominfo debian10
Id:             -
Name:           debian10
UUID:           a861f2ad-633d-440b-8d7e-c6b3499e021b
OS Type:        hvm
State:          shut off
CPU(s):         3
Max memory:     2097152 KiB
Used memory:    2097152 KiB
Persistent:     yes
Autostart:      disable
Managed save:   no
Security model: apparmor
Security DOI:   0
```

5) Découverte de l'architecture KVM

5.1) Gestion du réseau

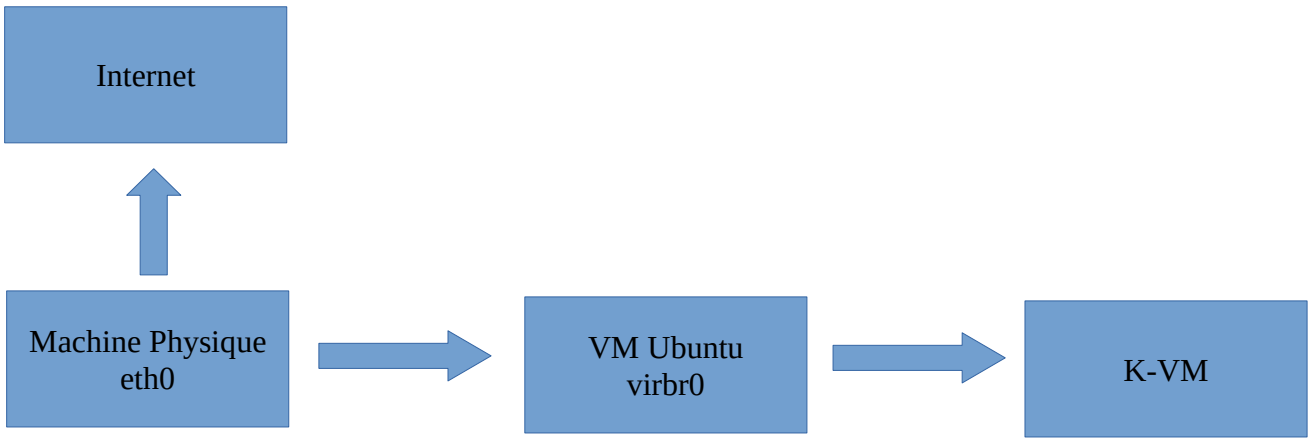
1)

```
virsh # net-info --network default
Name:           default
UUID:           2bf92031-28a6-48ad-a0b4-522191d24806
Active:         yes
Persistent:     yes
Autostart:      yes
Bridge:         virbr0
```

```
root@ubuntu:~# brctl show
bridge name     bridge id       STP enabled     interfaces
docker0         8000.024259c53f92       no
virbr0          8000.525400b18a0b       yes             virbr0-nic
                                        vnet0
```

Mon bridge de base est le virbr0

2)



3)

5.3) Clonage d’une K-VM et gestion du stockage

1)

2) J’éteins ma K-VM debian dans le shell virsh puis je fais la commande suivante pour cloner la K-VM :

```
root@ubuntu:~# virt-clone --original debian10 --name clone-debian10 --auto-clone
Allocation de « clone-debian10.qcow2 »
| 5.0 GB  00:00:02

Le clone « clone-debian10 » a été créé.
```