

TP Virtualization

Teabe Boris
boris.teabe@inp-toulouse.fr

The goal of this TP is to experiment with virtualization.

The goal of this labwork is to experiment with virtualization. On a big server, we used KVM to create VMs (running Ubuntu) which will be considered as your physical servers. Then, in these physical servers, you will install Xen (with its hypervisor) and then create a VM. Then, you will experiment with VM migration.

1. Installation

- pre-requisite
 - you should have a Linux system installed (provided by us in this TP)
 - preferably Ubuntu (I am working with ubuntu 20.04)
- We have already deployed the KVM and created for each group of students a VM that will be their server.

Use the following command to access your server:

ssh sepia@147.127.92.141 -p 130XX

where XX=01-40. This will give you access to a VM with IP address **192.168.27.(XX + 10)**.

2 Installation of Xen in your server

- install ssh in your server / **the ssh-server can already be installed if you use the VM we provided.**
 - sudo bash**
 - sudo apt-get install openssh-server -y**
- enable root login on your ssh server
 - edit **/etc/ssh/sshd_config (using vim or nano)**
 - configure
 - PermitRootLogin yes**
- reload your configuration
 - service ssh reload**
- provide a password for root
 - passwd root**
- install bridge tools
 - apt-get install -y bridge-utils**
- install Xen hypervisor
 - sudo apt-get install -y xen-hypervisor-amd64**
- update grub to boot the hypervisor and reboot
 - cd /etc/default/grub.d/**
 - modify the file « **xen.cfg** » and edit line **83** to set :

XEN_OVERRIDE_GRUB_DEFAULT=1

- and then update the grub

update-grub
reboot

- Reconnect to your working environment by redoing the ssh as at the beginning of the class.
- verify the Xen is working properly

sudo bash
xl list

- you should see one VM: the **dom0**

- update your network configuration to add a bridge
 - edit **/etc/netplan/50-cloud-init.yaml** (with **vim** or **nano**)
 - set this content :
 - you can type the command **"ip addr"** to get your IP address

network:

ethernets:

enp1s0:
dhcp4: no

bridges:

xenbr0:
interfaces: [enp1s0]
addresses: [<current IP address>/24]
gateway4: 192.168.27.1
nameservers:
addresses: [8.8.8.8]

version: 2

- Test your configuration
- netplan try**
- If everything is ok, press « Enter » to accept the new network configuration
 - verify that networking is available

ping 8.8.8.8
ping google.fr

3. Creation of a VM image

- Today, there are VM images available on linux distribution website. We will download a ubuntu image from the internet.

mkdir images
wget http://cloud-images.ubuntu.com/releases/focal/release/ubuntu-20.04-server-cloudimg-amd64.img -P /home/ubuntu/ -O vm.qcow2

- We need to create the VM configuration file

cat << EOF > /etc/xen/vm.cfg
bootloader = 'pygrub'
vcpus = 2
memory = 1024
root = '/dev/xvda1 ro'
disk = [

```

        '/home/ubuntu/vm.qcow2,qcow2,hda,rw'
    ]
    name      = 'myvm'
    vif = [ 'bridge=xenbr0' ]
EOF

```

- Modify the default password of the downloaded VM image

```

modprobe nbd max_part=8
qemu-nbd --connect=/dev/nbd0 /home/ubuntu/vm.qcow2
mount /dev/nbd0p1 /mnt/
chroot /mnt/

```

- Change the vm password with this command, and you should type the new password. **Don't forget the new password.**

```

passwd
exit

```

- Type the new root password for the VM and then.

```

umount /mnt/
qemu-nbd --disconnect /dev/nbd0
rmmod nbd

```

4. Creation of a VM

- create the VM

```

xl create /etc/xen/vm.cfg

```

- verify that the VM was created

```

xl list

```

- you should see your domU

- To connect to the console of the VM use the command, username root **password you mus know**

```

xl console myvm

```

- You need to configure the network in the VM (set the VM IP address add your IP address + 100 on the last number) as we did previously.

```

cat << EOF > /etc/netplan/50-cloud-init.yaml

```

```

network:

```

```

  ethernets:

```

```

    enp1s0:

```

```

      dhcp4: no

```

```

      addresses: [<current IP address + 100>/24]

```

```

      gateway4: 192.168.27.1

```

```

      nameservers:

```

```

        addresses: [8.8.8.8]

```

```

  version: 2

```

```

EOF

```

- Test your configuration

```

netplan try

```

- If everything is ok, press « Enter » to accept the new network configuration

- verify that networking is available

```

ping 8.8.8.8

```

ping google.fr

- **To disconnect from the console use this keyboard combinaison <Ctrl + shift + (>>**

5. Experiment with migration

- Select another group of students with whom you will carry out the migration : The group launching the migration will be consider as server1 and the other group as server2.

- to migrate **myvm** to **server2**, the filesystem of **myvm** (in vms) should be accessible in **server2**

- install NFS and mount the directory which includes the VM image

- in server1

sudo apt-get install nfs-server nfs-common

- edit **/etc/exports**

- add

/home/ubuntu/ *(rw,sync,no_root_squash)

- restart NFS

systemctl restart nfs-kernel-server

systemctl status nfs-kernel-server

- in server2

sudo apt-get install nfs-common

sudo mount -t nfs <IP address of server1>:/home/ubuntu/ /home/ubuntu

- verify that the VM image is accessible from server2

ls /home/ubuntu/

- you should see the same content as on server1

- in server1, trigger the migration

sudo xl migrate myvm <IP address of server2>

- verify that migration was performed

- in server1 and server2

sudo xl list

- migrate myvm back to server1

7. Experiment with liveness using ping command

- in server1, you can recover the console to myvm

sudo xl console myvm

- **To disconnect from the console use this keyboard combination <Ctrl + shift + (>>**

- From server1, we will use ping command to check that the vm is running during the migration

ping <myvm IP address>

- migrate myvm from server1 to server2

- the ping should not be interrupted

- destroy myvm