5e année ISS 2018/19

Cloud Computing: Adaptability and Autonomic Management

Lab 1 : Introduction to Cloud Hypervisors

S. Yangui (INSA/LAAS)

Access link : goo.gl/r5T3cX

Proxmox servers assignement : goo.gl/DDbdDt

Introduction

The general purpose of this lab is to enhance your knowledge of concepts and technologies for virtualization techniques. These techniques are used in IT environments that support the provisioning (i.e. deployment, management, etc.) of novel software systems (with their potential constraints such as QoS and security). These environments are typically dynamic and distributed.

The tutorials will be evaluated based on the work performed during lab hours, but also on your personal and dissertation efforts. The students must write an online synthesis in parallel during the lab hours. Involvement during lab hours will also be considered in the final mark.

In addition to the Web (just Google it!), we provide the following documentation:

- Cookbook 1 : *Proxmox yPBL cookbook*
  - http://bit.ly/yPBL_Proxmox_Cookbook
- Cookbook 2 : *Virtualization with Proxmox yPBL cookbook*
  - http://bit.ly/yPBL_proxmox_1_9_cookbook
- VirtualBox virtualization software manual
  - http://download.virtualbox.org/virtualbox/UserManual.pdf
- Proxmox hypervision software manual :
  - Wiki Proxmox : *http://pve.proxmox.com/wiki/Main_Page*
  - Documentation : *http://pve.proxmox.com/wiki/Documentation*
- *Container vs. KVM in Proxmox*
  - *http://pve.proxmox.com/wiki/Container_and_Full_Virtualization*

        - http://www.linux-kvm.org/page/Main_Page
    - *Proxmox command line*
        - http://pve.proxmox.com/wiki/Command_line_tools
    - *Virtual appliance library*
        - http://www.turnkeylinux.org/
        - http://pve.proxmox.com/wiki/Category:Virtual_Appliances

Lab objectives

The purpose of this lab is to learn the basic functionalities (features) that cloud service providers could offer to end-users at the IaaS level. The lab objectives are detailed in what follows:

- Objective 1: Knowing the difference between the types of hypervisors' architectures (type 1/type 2), being able to classify the hypervisors used in the labs (i.e. VirtualBox and Proxmox), and being able to use their basic functionalities.
- Objective 2: Knowing the difference between the two main types of *virtualisation hosts*[1], i.e. *virtual machines* (VM) and *containers* (CT), as well as, the different container types.
- Objective 3: Knowing the difference between the modes of network connection for VM/CT (i.e. NAT and Bridge modes)
- Objective 4: Being able to create VirtualBox VMs, in NAT mode, and make configurations so that they can access and be accessed through the Internet
- Objective 5: Being able to create Proxmox CT, in bridge mode (via a DHCP server)
- Objective 6: Being able to manually snapshot and restore a Proxmox CT
- Objective 7 : Being able to manually migrate a CT in Proxmox

Lab format

The lab is organized in two main parts:

1. A theoretical part that you need to start with. You need to carefully read and assimilate to recall all the concepts introduced during the lecture.
2. A practical (hands-on) part. During this work, the students need to write a minutes in an online shared document (1 link per lab duo).

[1] According to the ETSI terminology.

Theoretical part
(objectives 1 to 3)

1. Difference between the main virtualisation hosts (VM et CT)

There are two types of virtualisation hosts (i.e. VMs and CTs). These hosts are depicted in Figure 1.

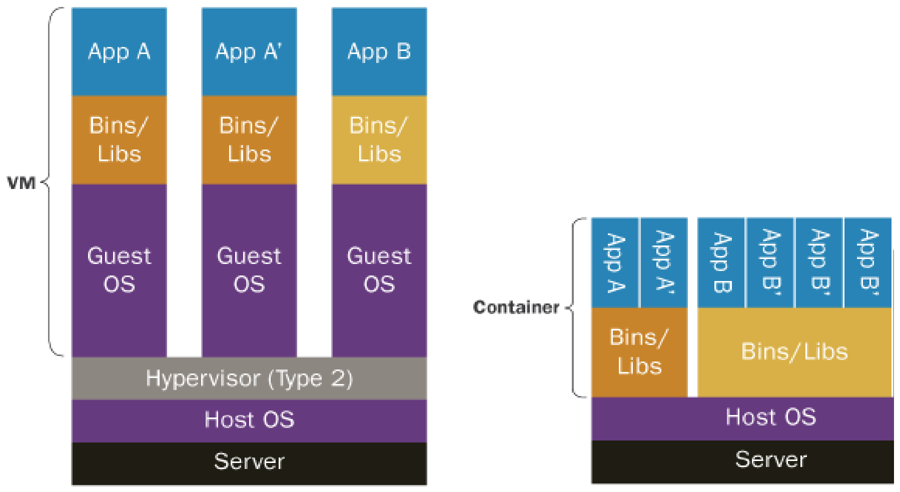


Figure 1: VM vs CT

Tasks (You need to write down your answers in the shared document):

- Understand the figure above, and discuss it.
- Compare the two type of hosts based on two perspectives: from an application developer's point of view, and from an infrastructure administrator point of view. For each one of these actors, the comparison should be based on the following criteria:
    - virtualization cost, taking into consideration memory size and CPU,
    - Usage of CPU, memory and network for a given application,
    - Security for the application, regarding access right and resources sharing,
    - Performances regarding response time,
    - Tooling for the continuous integration support

2. Differences between the existing CT types

Different CT technologies are available (e.g. LXC/LXD, Docker, Rocket, OpenVZ, runC, containerd, systemd-nspawn). Their respective positioning is not obvious, but comparative analyses are available on the Web, such as the one displayed in Figure 2.

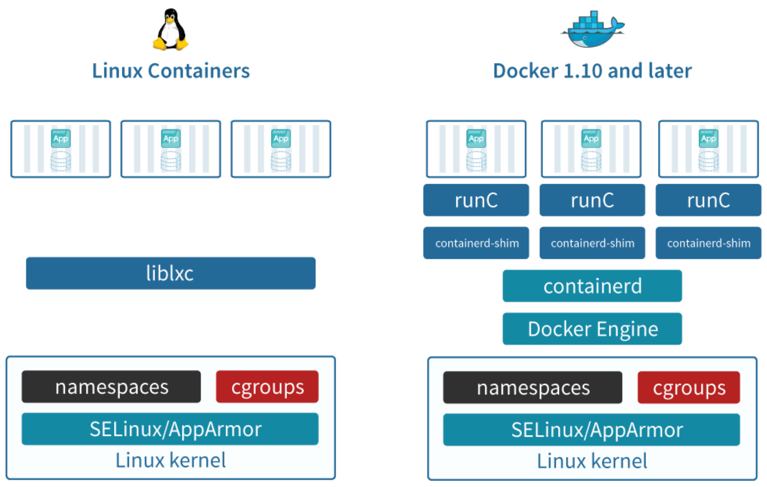


Figure 2 : Linux Lxc vs Docker

These technologies can however be compared based on the following criteria (non-exhaustive list):

- Application isolation and resources, from a multi-tenancy point of view,
- Containerization level (e.g. operating system, application),
- Tooling (e.g. API, continuous integration, or service composition).

Tasks (You need to write down your answers in the shared document):

- Elaborate on the proposed criteria,
- From existing Web resources, classify the existing CT technologies (LXC, Docker, Rocket, …) based on these criteria, and eventually, on additional criteria that you need to identify by yourself.

3. Differences between Type 1 & Type 2 for hypervisors' architectures

There are two main categories of hypervisors, referred to as type 1 and type 2.

Tasks (You need to write down your answers in the shared document):

- Based on the lecture's slides (slides 14, 15 and 16), elaborate ob each of these categories.
- Identify to which architecture type VirtualBox and Proxmox belong to.

4. Difference between the two main network connection modes for virtualization hosts

Tasks (No writing required)

With some hypervisors, multiple possibilities exist to connect a VM/CT to the Internet, via the host machine (in this case your PC). The two main modes are the following:

- *NAT mode*: It is the default mode. It does not require any particular configuration. In this mode, the VM/CT is connected to a private IP network (i.e. a private address), and uses a virtual router (managed by the hypervisor) running on the host machine to communicate outside of the network:
    - This router executes what is equivalent to a NAT function (only *postrouting*) allowing the VM to reach the host machine or the outside;
    - However, the VM cannot be reached from the host (and by any another VM hosted on the same machine): to make it accessible from outside, it is necessary to deploy port forwarding (*prerouting*) in VirtualBox.
- *Bridge mode*, the most widely used (yet not systematically), in which the VM/CT sees itself virtually connected to the local network of its host (see Figure 3): it has an IP address identifying it on the host network, and can access the Internet (or is accessed) as the host.

NB: Other modes exist in VirtualBox, such as Private Network, that you can try out.

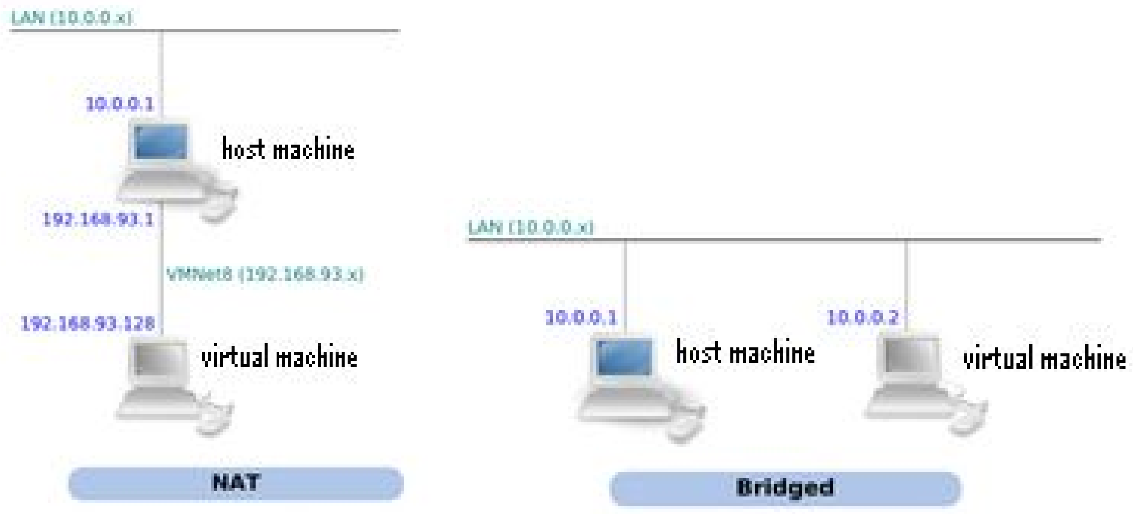


Figure 3 : Mode NAT vs Mode Bridge (the most usual)

Practical part
(objectives 4 to 7)

For each item thereafter, you must make a short written report on the shared document.

You must apply the following naming rules for the VM and CT you create:

- Creation of the **VM** n°**4** by duo **3** of group **A**
  - **vm**-tpiss-virt-**A3**-vm**4**
- Creation of the **CT** n°**3** by duo **2** of group **B**
  - **ct**-tpiss-virt-**B2**-ct**3**

1. Tasks related to objectives 3 and 4

Creating a VirtualBox VM (in NAT mode), and setting up the network to enable two-way communication with the outside

In this part, you will use the VirtualBox hypervisor in NAT mode to connect a VM to the network. The bridge mode will be used in the second part of the lab, with another hypervisor (Proxmox).

Taks:

First part: Creating and configuring a VM

- Copy the virtual hard drive (vm-alpine.vmdk.zip)
- Unzip the archives.
- Start VirtualBox (available on Windows sessions).
- In VirtualBox, create a VM - Type Linux Other Linux 64 bits (name it according to the previously provided naming rules) and configure it:
  - You need to associate the following resources to it: RAM (for instance 512 Mo), hard drive (provided) and CPU (1 core max),
  - Associate peripherals (e.g. network board, CD readers, USB ports), choosing a NAT mode network board (the private IP address is automatically attributed).
- Run the VM. If necessary, authentication information are the following:
  - username : **user**
  - password: **user**
  - root password : **root**

Second part: Testing connectivity

Identify the IP address attributed to the VM using ifconfig (in Linux command line), and compare it to the host address (ip config in the command line). What do you observe.

With a ping command (or any other command of your choice) :

- Check the connectivity from the VM to the outside. What do you observe?
- Check the connectivity from your neighbour's host to your hosted VM. What do you observe?
- Check the connectivity from your host to the hosted VM. What do you observe?

VM duplication

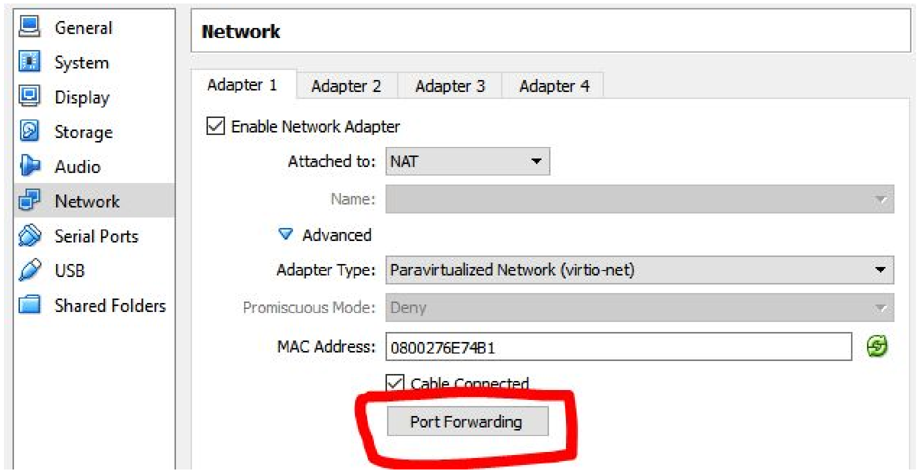
To create a new (clone) VM with the same disk file:

- Create a copy of fichier vm-alpine.vmdk → vm-alpine-copy.vmdk
- Run the following command:
    - **VBoxManage internalcommands sethduuid chemin\vers\vm-alpine-copy.vmdk**
    - *NB: VBoxManage is not declared globally. It must be invoked with C:\Program Files\Oracle\VirtualBox\VBoxManage.exe in Windows.*
- Create the new VM with the vm-alpine-copy.vmdk file

NB: The original copy of the file is not sufficient , because it embeds the same identifier as the original, which must be unique in the hypervisors namespace. VirtualBox will report an error.

Third part: Set up the "missing" connectivity

Via the *Port Forwarding* technique, fix the issues identified in the previous part, for a dedicated application (e.g. SSH, port 22). To that end, you need to add a new forwarding rule to the management interface of the network board provided by VirtualBox,
NB : On Windows, you can use PuTTy as a SSH client.



☑ Enable Network Adapter
Attached to: NAT
Name:
Adapter Type: Paravirtualized Network (virtio-net)
Promiscuous Mode: Deny
MAC Address: 0800276E74B1
☑ Cable Connected

Figure 4 : Port forwarding

2. Expected work for objectives 5, 6 and 7

Create a Proxmox CT (in Bridge and DHCP mode)
Manually snapshot and restore this CT
Manually migrate this CT from the host server to another

Tasks :

First part : CT creation and configuration

- Connect to the Proxmox Web interface: https://srv-px1.insa-toulouse.fr:8006/ (accept the security exception for the SSL certificate)
  - *NB : If you are connecting from outside INSA, the use of INSA VPN is required.*
- Authenticate with your INSA login/pwd using the groupe "**Ldap-INSA**"
  - Close the notification about the Proxmox license.
- For all the manipulations, you must use the server attributed to your duo (link at the top of the document).
- Create a CT (in the resource pool **"admin"**) based on the available template **debian-8-turnkey-nodejs_14.2-1_amd64.tar.gz**, and configure it:
  - Set up a password
  - Ressources : Disk space (not exceeding **3** Go, on the **"vm"[nfs]** reader), RAM (not exceeding **512** Mo), CPU (**1** core)
  - Bridge mode network (**vmbr1** and **VLAN ID = 2028**) with dynamic IP addresses (**dhcp**)
  - Go to the CT view (via the Pool View or the Server view)
  - Start the CT, and activate the Console
  - Follow the "TurnKey" initial configuration steps (login: **root** ; password: the one you defined).
    - NB: On the first start up, you can skip the three first TurnKey blue screens

Second part: Connectivity test

Identify the IP address attributed to the container via ifconfig and compare it to the address of the Proxmox server hosting the CT. Comment, with respect to the presentation of the NAT and Bridge modes done in the theoretical part.

Using a ping command (or a command of your choice) :

- Check the connectivity from the created CT to a Proxmox server (see Figure 4). What do you observe.
- Check the connectivity from your own machine to the created CT.

Third part : Snapshot and restoring a CT

Go to the Backup item of the CT view:

- Snapshot the CT with the “backup now” option
- Remove a software from the CT, for instance “nano” (**apt-get remove nano**)
- Restore the CT from the snapshot.

What do you observe (regarding the presence of nano)?

Fourth part: CT migration

- Migrate the CT from a server to another. What do you observe?