CAP-GIA (FIB) Laboratory Assignment

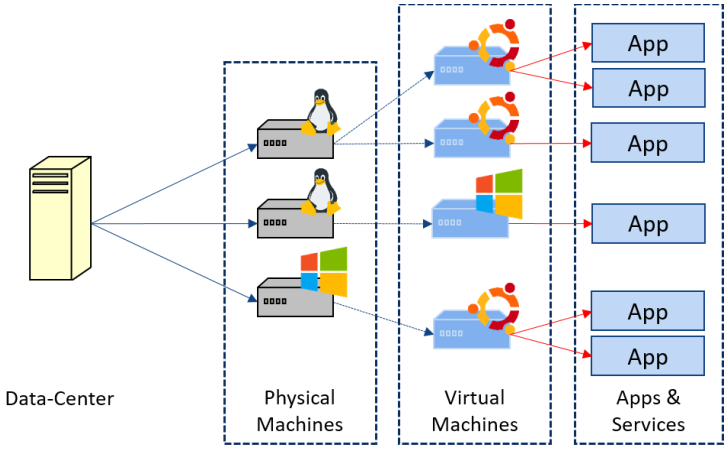
Lab 1: Màquines Virtuals

Jordi Torres and Josep Lluís Berral

Fall 2024

Màquines Virtuals

La virtualització ens permet poder recrear computadors (des del hardware fins a les aplicacions, passant pel sistema operatiu) dins un procés com una aplicació més. Això ens dona la capacitat de desplegar entorns complets sense haver d'instal·lar-los directament en una màquina física, podent tenir múltiples màquines amb diferents sistemes funcionant alhora. Mitjançant Màquines Virtuals (VMs) podem executar un entorn isolat de la resta del sistema, i córrer serveis i aplicacions de forma separada d'altres usuaris i VMs. A més, les VMs es poden aturar per tal de continuar més tard, o transportar i copiar a altres màquines físiques, per tal de ser migrades o replicades. Una sola imatge de VM pot ser distribuïda a través de la xarxa per a que diferents usuaris o sistemes puguin engegar instàncies pròpies. Això ens permet tenir sistemes sencers "auto-continguts" dins una VM, i compartir o escalar sistemes amb facilitat. Aquesta sessió estarà centrada en el desplegament i experimentació amb màquines virtuals, usant l'hypervisor de virtualització com a "hands-on" de laboratori.



VirtualBox és un hypervisor encarregat de virtualitzar instàncies de VMs. En aquest laboratori veurem com obtenir imatges pre-definides a partir d'un repositori d'imatges de VM, i crear imatges pròpies. En aquest laboratori treballem amb **Debian 12 (64bits)** on tindrem VirtualBox i Vagrant, i veurem els processos de sistema que tenen relació amb la virtualització gràcies a les eines de Unix/Linux.

Objectius del Laboratori

1. Connectar-se a un sistema de còmput remot usant SSH
2. Crear, configurar, iniciar, usar i apagar una màquina virtual
3. Crear una xarxa virtual, amb més d'una màquina virtual connectada
4. Instal·lar un servei a la VM, i que quedi exposat a l'exterior (e.g. un servei web)

Nota important: Compte amb fer "copy & paste", ja que hi ha símbols com comes o guions que es poden copiar malament d'aquest document a consola. Si no us va una comanda, escriviu-la directament!

Índex de continguts

Instruccions de la pràctica

Les Màquines Virtuals requereixen desplegar-se sobre una màquina "amb privilegis", i per tant no ho podem executar a un clúster clàssic (e.g., Boada o MareNostrum). En canvi, a màquines al Cloud disposem d'aquests privilegis. Aquesta pràctica la farem a unes màquines preparades per a vosaltres.

Les màquines preparades (Clúster CROMAI-CAP) disposen de **Debian 12**, i seran **compartides per tots els estudiants**. Això vol dir que haurem de ser conscients que compartim recursos limitats. Aquestes màquines tenen instal·lat VirtualBox (hypervisor) i Vagrant (gestió d'imatges de VMs).

Grups de Treball

Per a aquesta pràctica ens distribuïm en parelles (**grups de 2 persones**). Cada parella tindrà un usuari i una màquina assignada a *cromai-cap*. Cada parella farà una sola entrega; una recomanació és que una persona del grup faci el rol d'executor, i l'altre membre del grup faci de secretari, i anar anotant les respostes del qüestionari que cal anar responent.

Qüestionari de Laboratori

Durant la pràctica trobarem preguntes que cal anar contestant a mesura que avanceu per la pràctica. Aquestes formen el **qüestionari de laboratori** que ens servirà de guia durant el laboratori.

Disposareu de la sessió de laboratori per a fer la pràctica, i d'una setmana per a **entregar el qüestionari pel Racó**, on veureu una pràctica oberta per a "Màquines Virtuals". Les respostes les entregareu usant el document plantilla de la pràctica, indicant les respostes sota les corresponents preguntes. Assegureu-vos de justificar bé les respostes, i suposeu al voltant d'una línia o dos per resposta, tampoc més. No enganxeu "screenshots". Eviteu entregar al darrer moment, per evitar possibles problemes tècnics, i per a que us pugueu assegurar que heu pujat la pràctica correctament.

Temps de Dedicació

Aquesta pràctica ha estat dissenyada per a que la pugueu realitzar en **aproximadament dues hores**. Hem tingut en compte el que es triga en descarregar el software i les imatges de VM.

Avaluació

L'avaluació de la sessió presencial tindrà en compte l'assistència i participació a classe, així com el qüestionari entregat.

Nota Qüestionari → Suma de les puntuacions per cada pregunta.

1. Entorn Cloud i Hypervisor

L'hypervisor es el software que ens permet executar la virtualització d'una màquina com si fos una aplicació més. Ens permet encapsular en un procés tot un sistema operatiu, software i llibreries, a més de crear una "xarxa virtual" per tal tenir una xarxa privada que connecti totes les VMs, i a més de crear un pont ("bridge") entre la xarxa virtual i els dispositius físics de xarxa de la màquina "hoste" per a que les VMs puguin accedir a la xarxa global (e.g. Internet o a la màquina hoste).

En aquesta pràctica usarem VirtualBox com a "hypervisor", i usarem Vagrant com a gestor automàtic de VMs. Podem crear, configurar i gestionar VMs de forma manual amb comandes del virtualitzador corresponent, però existeixen gestors de VMs com ara Vagrant que automatitzen la creació, configuració i gestió de VMs amb comandes bàsiques. Primer, entrarem a la màquina *cromai-cap* assignada, amb el *user* i *password* donats:

```
ssh <usuariYYY>@<màquina assignada>        # SSH: Port 22 per defecte
```

Ara teniu accés a l'hypervisor i a un gestor d'imatges preparats per fer servir.

Compartir recursos

Com a detall important, les màquines *cromai-fog00* i *cromai-fog01* disposen de:

```
CROMAI-FOG00:

 CPUs:           12 Threads
 Memòria:        48 GB

CROMAI-FOG01:

 CPUs:           12 Threads
 Memòria:        24 GB
```

DISCUSSIÓ: Si som 48 estudiants, en grups de 2 persones (24 grups), repartits en 2 subgrups de laboratori (12 grups per cada laboratori), i l'exercici demana que engeguem 2 VMs de [1 CPU; 1 GB RAM]. Quins problemes podríem arribar a tenir?

Important: L'entorn per a fer la pràctica NO TÉ proteccions de quota de recursos (a propòsit), tal que si algú s'apropia de més recursos dels assignats, això afectarà a tots els companys. Recordeu d'apagar les VMs un cop heu fet la feina, i així allibereu els recursos.

2. Repositori d'Imatges de VM

Creant una VM pre-definida amb Ubuntu

Un cop disposem d'un sistema amb recursos, hypervisor i gestor d'imatges, podem descarregar una VM amb (e.g.) Ubuntu pre-instal·lat. A la web de Vagrant (https://app.vagrantup.com/boxes/search) podem veure la llista d'imatges ja pre-creades que podem usar. En el nostre cas usarem la imatge "ubuntu/jammy64", disponible per a VirtualBox. Primer, creem un directori on desar el **descriptor de VM**, i fem servir la comanda "init" per crear el fitxer "Vagrantfile" ja configurat per a "ubuntu/jammy64". Les següents comandes funcionen tant per Linux, MacOs com Windows, des de la consola de comandes:

```
mkdir $HOME/vm_cap                # Crear el directori de treball
cd $HOME/vm_cap                   # Entrar al directori
vagrant init ubuntu/jammy64       # Descarregar el Descriptor de VM
```

Ara hem de modificar el descriptor per, p.e., donar-li més memòria RAM, més CPUs, o donar-li un "nom". Obrim el fitxer "Vagrantfile" amb un editor de text (p.e. vim o nano), i afegim les següents línies **entre els tags de "config"**:

$HOME/vm_cap/Vagrantfile

```
  config.vm.provider "virtualbox" do |vb|
        vb.memory = "1024"
        vb.cpus = "1"
        vb.name = "VM1"
  end
```

Podem veure dins el descriptor de VM que ja tenim introduït per defecte "ubuntu/jammy64". De fet, si volguéssim introduir una VM d'un altre repositori o d'una VM exportada o una VM local, podem indicar-ho dins del tag de "config":

$HOME/vm_test/Vagrantfile (EXEMPLE)

```
  config.vm.box = "ubuntu/jammy64"
# config.vm.box_url = "<URL-DEL-REPO | URI-DEL-FITXER-VBOX>"
```

Amb això li hem donat a la màquina accés a 1GB de RAM, accés a 1 CPU, i de nom "VM1". Hem de tenir en compte que la nostra màquina física ("hoste") ha de disposar d'aquests recursos per a poder assignar-los a la VM. També hem de tenir en compte de no donar tots els recursos de l'hoste a les VMs, ja que el sistema operatiu hoste i el software de l'hypervisor necessiten recursos per funcionar. Un cop configurada la VM, podem iniciar-la. El primer cop serà descarregada dels repositoris de Vagrant i muntada ("aprovisionada"), i llavors s'iniciarà:

```
vagrant up                        # Provisiona (1er cop) i Inicia la VM
```

EXERCICI 1: Veient la traça d'inici de "vagrant up":

a) Quina informació ens proporciona l'output (un cop aprovisionada) respecte a SSH, xarxa virtual, ports oberts, i directoris compartits entre la VM i l'hoste?

Si examinem els processos corrents a la nostra màquina hoste, veurem que tenim els processos de l'hypervisor, el procés de la màquina virtual que hem engegat, i el procés que ens crea una xarxa privada virtual. Si fem un "ps" podem veure aquests processos:

```
ps aux | grep "virtualbox"
```

Veiem que tenim processos de "VBox" (l'hypervisor), un procés de "VBoxHeadless" (la màquina virtual, corrent sense interfície gràfica en aquest cas).

EXERCICI 2: Veient la informació del "ps":

b) Quina informació podem extreure de la VM?
c) Quins processos relacionats apareixen a més a més de la VM en execució? I què fan?

Examinant els fitxers de la VM

Donat que Vagrant + VirtualBox desen les dades de la VM a "$HOME/VirtualBox VMs", busqueu el directori amb la informació de "vm_cap".

EXERCICI 3: Llistant i examinant els fitxers de la màquina virtual:

d) Quins fitxers trobem, quins son els que podem considerar importants, i què fan cadascun?
e) Si volguéssim migrar aquesta VM a un sistema Cloud, quins fitxers ens caldria moure?

Entrant a la VM

Ara entrarem a la VM. Quan hem fet "vagrant up" hem vist que Vagrant ja ens ha obert a la VM un port per connectar-nos per SSH:

```
==> default: Forwarding ports...
    default: 22 (guest) => XXXX (host) (adapter 1)
    ...
    default: SSH address: 127.0.0.1:XXXX
    default: SSH username: vagrant
    default: SSH auth method: private key
```

Això ens indica que podem connectar-nos per SSH a la màquina un cop engegada, usant el port XXXX de la màquina hoste (connectat al port 22 d'SSH de la VM). Com que estem en una màquina compartida, cada VM creada rebrà un port diferent (de la màquina física, XXXX) i el connectarà al port de SSH (de la màquina virtual, 22). D'aquesta forma, ens podríem connectar a la nostra VM usant el usuari per defecte (*vagrant*:*vagrant*), fent :

```
ssh vagrant@127.0.0.1 -P XXXX   # Obrir una sessió de Secure-Shell
```

O podem usar la següent comanda, molt més còmoda, que ens automatitza l'accés:

```
vagrant ssh                     # Obrir una sessió de Secure-Shell
```

Per sortir només hem de fer “exit”, i per aturar-la podem fer “vagrant halt” des de fora. Per iniciar-la de nou, només ens cal fer “vagrant up”. Si volem esborra la VM, podem fer “vagrant destroy”, i fent “vagrant up” després tornarem a descarregar la imatge nova. Un cop engegada, podem tornar a fer “vagrant ssh”.

3. Exposar la VM a la Xarxa

Preparació d’un servei

Dins de la VM instal·larem el servidor web NGINX, per servir pàgines des de la VM:

```
sudo apt-get update
sudo apt-get install nginx
```

Quan estem a la VM podem accedir a la xarxa, ja que l’hypervisor ens prepara per defecte NAT (la VM pot sortir a Internet directament a través de la interfície de xarxa de la màquina hoste, usant ports per distingir entre el tràfic de l’hoste i el de la VM). No només hem pogut descarregar NGINX usant el gestor de paquets d’Ubuntu, sinó que podem navegar per la web usant el navegador “lynx”:

```
lynx https://www.fib.upc.edu                    # Podeu sortir amb "q"
```

Si no tenim la comanda “lynx” a la VM, la podem instal·lar amb “apt-get” tal com ho hem fet amb “nginx”. Si fem “lynx http://localhost” veurem la pàgina d’inici per defecte de NGINX. Veiem que el servidor web està en funcionament. Si ara sortim de la VM, veiem que no podem accedir a NGINX ja que està dins la VM, i aquesta està “tancada a l’exterior”.

Ara modificarem la configuració de la VM per exposar el port d’NGINX a l’exterior, i poder servir web des de la VM. Tancarem la VM fent “vagrant halt”, i Vagrant intentarà tancar-la de forma “educada” (enviant una ordre de shut-down, en comptes de matar el procés virtualitzat de bones a primeres). Un cop la VM estigui apagada, modifiquem el “Vagrantfile” i afegim una redirecció de ports:

$HOME/vm_test/Vagrantfile

```
  config.vm.network "forwarded_port", guest: 80, host: 8YYY
```

Com que estem compartint la màquina amb molts més usuaris, hem de buscar un número de port únic per a nosaltres. Per a tenir un número únic, li donarem `8YYY` on YYY = el número de grup assignat. Al tornar a fer “vagrant up” veiem que ens apareix un missatge nou a la seqüència d’inici:

```
==> default: Forwarding ports...
    default: 80 (guest) => 8YYY (host) (adapter 1)
```

Això ens indica que podem connectar-nos al servei web que hem instal·lat, usant el port `8YYY` (connectat al port 80 d’NGINX de la VM). Si ara fem “lynx http://localhost:8YYY”, podem veure que NGINX de la VM respon.

Amb les comandes de control de VirtualBox podem visualitzar la informació de la VM (a més de controlar-la o modificar-la manualment, sense Vagrant):

```
vboxmanage showvminfo VM1
```

4. Creant una Xarxa Virtual

Creem una Xarxa Virtual

La xarxa de la VM ara ens ho gestiona tot l'hypervisor VirtualBox com a NAT per defecte. El que podem crear és una Xarxa Virtual, que ens permeti engegar més VMs, connectar-les entre elles, i que en comptes d'exposar ports al hoste, tenir-los disponibles a la xarxa virtual (per més tard encaminar-los, p.e., amb IPTables o altres sistemes de redirigir tràfic de xarxa).

Apaguem la VM per fer modificacions ("vagrant halt"). Obrim el "Vagrantfile", comentem la línia de "port forwarding", i afegim aquesta línia dins el tag de "config", per indicar que volem una xarxa virtual: i quina IP volem que tingui la VM:

$HOME/vm_test/Vagrantfile

```
config.vm.network "private_network", ip: "10.YYY.0.101"
```

Tornem a iniciar la VM ("vagrant up"). Ara veurem que tenim 2 adaptadors de xarxa a la VM, un amb NAT i un amb la Xarxa Virtual:

```
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
    default: Adapter 2: hostonly
```

EXERCICI 4: Comprovem les IPs tant a l'hoste com a la VM:

f) Com està l'hoste connectat a la Xarxa Virtual? Quina IP ha rebut?

Per veure que ara podem accedir directament a la VM sense traslladar ports, fem "lynx http://10.YYY.0.101" (sense indicar ports, ja que el 80 és el port per defecte en navegadors web!), i veiem que trobem directament NGINX sense exposar el port. Ara l'hoste i la VM es poden veure directament usant les respectives IPs.

Creem una nova VM

Ara, deixem la VM1 funcionant (sortim de la sessió però no l'apaguem). Per crear una nova VM, iniciem un nou directori, i fem "vagrant init" de nou. Compte amb copiar Vagrantfiles, ja que de vegades dona problemes (Vagrant detecta que el descriptor de VM "s'ha mogut", i intenta relacionar les VMs en comptes de crear-ne una de nova).

```
mkdir $HOME/vm_cap2              # Crear el directori de treball
cd $HOME/vm_cap2                 # Entrar al directori
vagrant init ubuntu/jammy64      # Descarregar el Descriptor de VM
```

Obrim el fitxer "Vagrantfile" per tal de definit la VM, donant-li un nom nou, i indicant-li que estarà a la xarxa virtual amb una nova IP:

$HOME/vm_test/Vagrantfile

```
config.vm.network "private_network", ip: "10.YYY.0.102"
config.vm.provider "virtualbox" do |vb|
      vb.memory = "1024"
      vb.cpus = "1"
      vb.name = "VM2"
end
```

Ara, iniciem la VM nova amb "vagrant up", i ens fixem en la traça d'inici.

EXERCICI 5: Veient la traça de desplegament ("provision"):

g) Ara tenim 2 VMs en marxa, amb els ports de SSH disponibles, però donats a l'atzar. Què hauríem de fer per a donar-los a cadascun una redirecció del port SSH com a 2YYY (VM1) i 3YYY (VM2)?
h) Si entrem a la VM nova, quina IP rep? Pot veure a la VM1, i com ho podem saber?

Final de la Pràctica

Tanqueu totes les VMs que tingueu en marxa. Responeu les PREGUNTES de forma breu, convertiu el document a **PDF**, i finalment **entregueu el qüestionari** (un per parella).