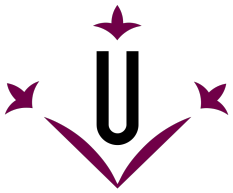

Universitat de Lleida

Virtualització

Seguretat d'Aplicacions i Comunicacions - Grau en Enginyeria Informàtica

Pablo Fraile Alonso

7 de novembre de 2022

Índex

Índex de figures

1 Configuració de la xarxa a VirtualBox

Ens demanen el disseny de xarxa de la figura 1.

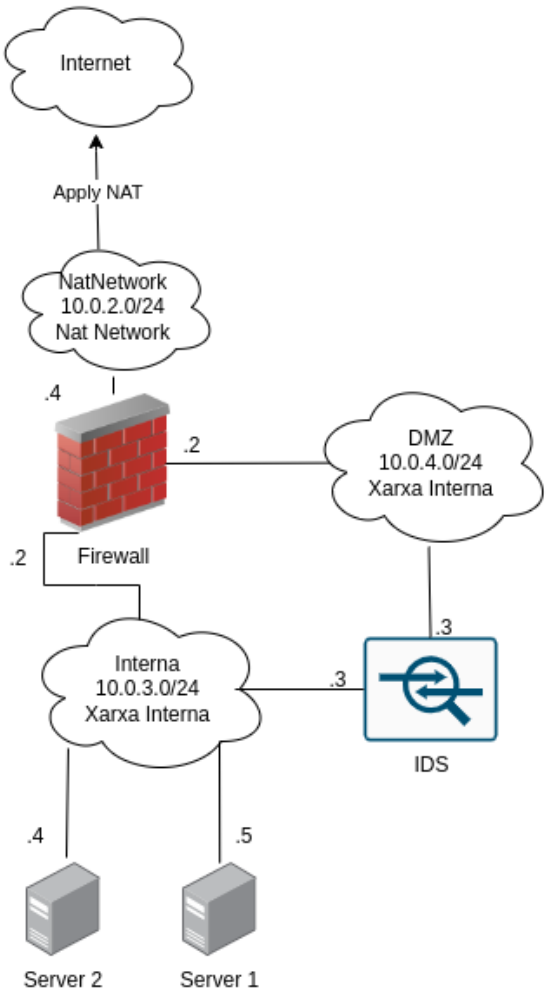

Figura 1: Disseny de la xarxa a virtualbox

Algunes justificacions són:

1. S'han emprat xarxes internes per a la xarxa *DMZ* i la xarxa *interna* ja que l'única necessitat que tenien era poder tindre comunicacions entre les VM's de la mateixa xarxa.

2. S'ha emprat una Nat Network per permetre connectar-nos a internet. Es podria haver emprat NAT o bridge, però s'ha decidit NAT-Network per tal de que, si qualsevol dia es volgués afegir una altra xarxa adjacent, es poguessin comunicar entre elles.

3. Com no hi ha cap servidor DHCP a les xarxes, s'han d'assignar les rutes per defecte i les ip's/màscares de forma manual. Les comandes són:

```
$ ip addr add <ip>/<mask> dev <interface>
$ ip route add default via <ip gateway>
```

1.1 Configuració de xarxa dels equips

1.1.1 FireWall

La configuració de xarxa del firewall es pot veure a la figura 2. La configuració de les interfícies a Cent-OS es pot veure a la figura 3.

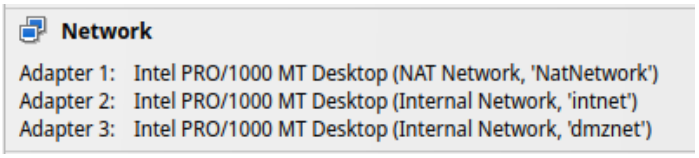

Figura 2: Configuració de les interfícies a la VM firewall

1.1.2 IDS

La configuració de xarxa del IDS es pot veure a la figura 4. La configuració de les interfícies a Cent-OS es pot veure a la figura 5.

1.1.3 Servers

La configuració de xarxa dels servidors (és la mateixa per als dos) es pot veure a la figura 6. La configuració de les interfícies a Cent-OS es pot veure a la figura 7 per al servidor 1 i a la figura 8 per al servidor 2.

```
[root@localhost ~]# ifconfig -a
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.4  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::6c18:10c2:8040:259c  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:1a:ad:15  txqueuelen 1000  (Ethernet)
        RX packets 68  bytes 7453 (7.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 87  bytes 7913 (7.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.3.2  netmask 255.255.255.0  broadcast 0.0.0.0
        ether 08:00:27:92:5e:ee  txqueuelen 1000  (Ethernet)
        RX packets 133  bytes 34976 (34.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 14  bytes 1144 (1.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s9: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.4.2  netmask 255.255.255.0  broadcast 0.0.0.0
        ether 08:00:27:36:3c:c8  txqueuelen 1000  (Ethernet)
        RX packets 4  bytes 316 (316.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 316 (316.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[root@localhost ~]# firewall_
```

Figura 3: ifconfig a la VM firewall

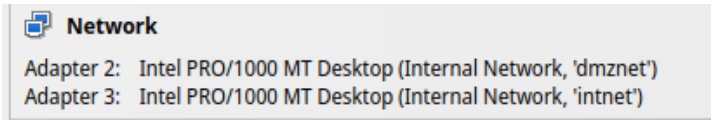

Figura 4: Configuració de les interfícies a la VM IDS

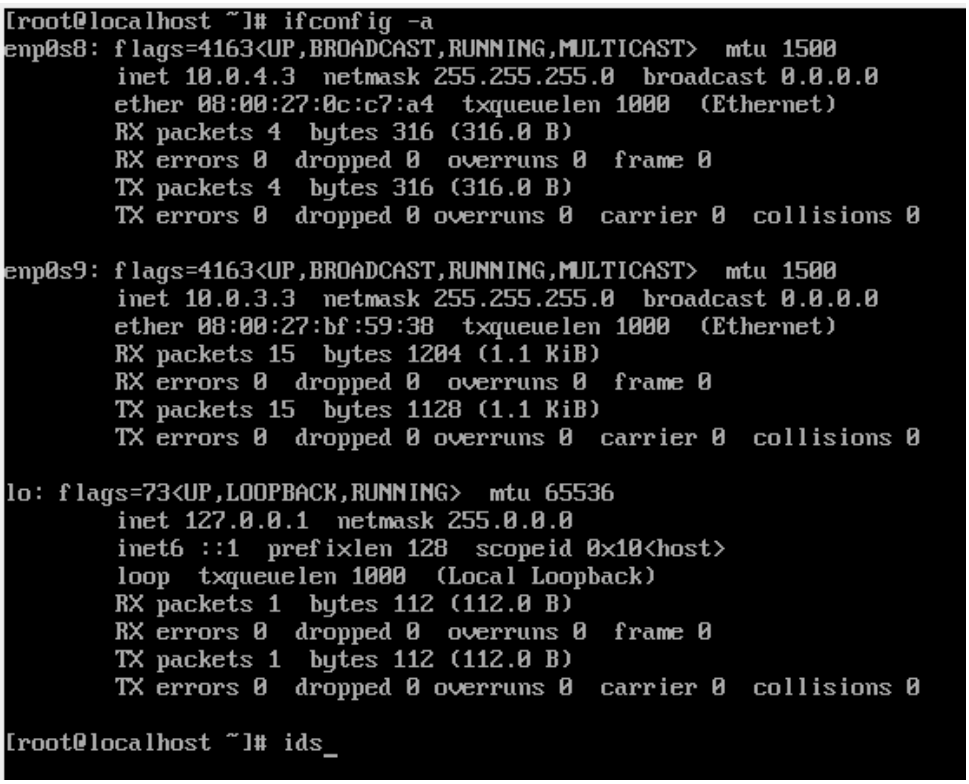

Figura 5: ifconfig a la VM IDS

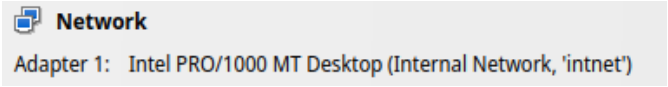

Figura 6: Configuració de les interfícies a les VM's servidors

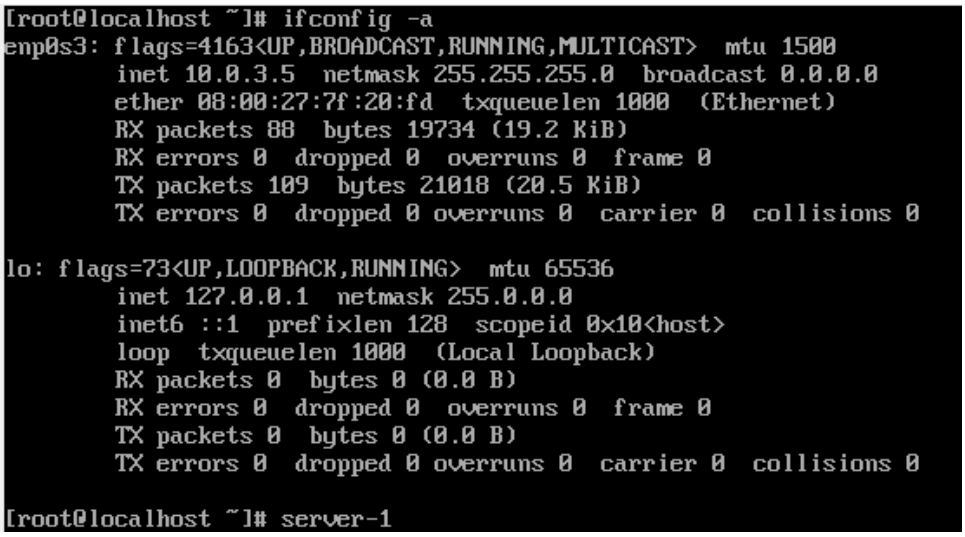

Figura 7: ifconfig a la VM server-1

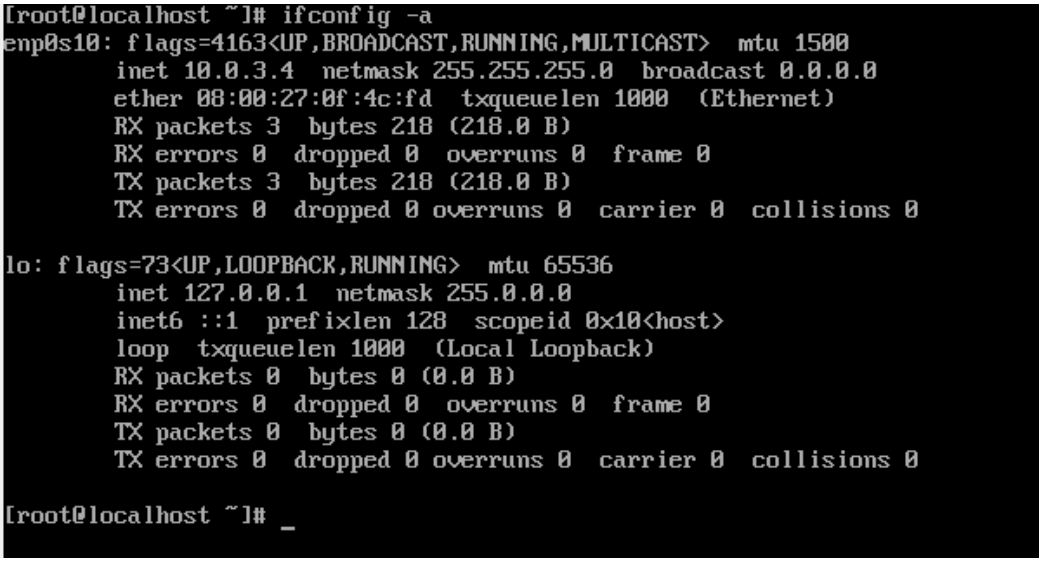

Figura 8: ifconfig a la VM server-2

1.1.4 Configuració de la xarxa Nat-Network

La configuració de la xarxa Nat-Network és la de la figura 9. S'ha deshabilitat dhcp per tal de saber sempre quina ip té el firewall.

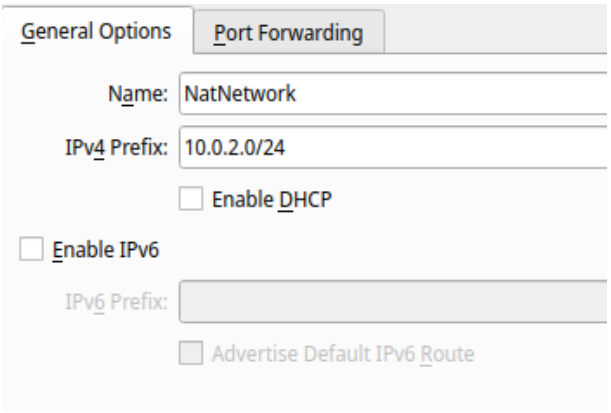

Figura 9: Configuració de la xarxa Nat-Network

1.1.5 Comprovació de la connexió entre els equips

La comprovació de connexió entre els equips s'ha fet a partir del firewall, on s'ha fet pings a tots els dispositius i a internet (figura 10).

```
[root@localhost ~]# ping -c 1 10.0.3.3
PING 10.0.3.3 (10.0.3.3) 56(84) bytes of data.
64 bytes from 10.0.3.3: icmp_seq=1 ttl=64 time=0.642 ms

--- 10.0.3.3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.642/0.642/0.642/0.000 ms
[root@localhost ~]# ping -c 1 10.0.3.4
PING 10.0.3.4 (10.0.3.4) 56(84) bytes of data.
64 bytes from 10.0.3.4: icmp_seq=1 ttl=64 time=2.25 ms

--- 10.0.3.4 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.251/2.251/2.251/0.000 ms
[root@localhost ~]# ping -c 1 10.0.3.5
PING 10.0.3.5 (10.0.3.5) 56(84) bytes of data.
64 bytes from 10.0.3.5: icmp_seq=1 ttl=64 time=1.26 ms

--- 10.0.3.5 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.255/1.255/1.255/0.000 ms
[root@localhost ~]# ping -c 1 10.0.4.3
PING 10.0.4.3 (10.0.4.3) 56(84) bytes of data.
64 bytes from 10.0.4.3: icmp_seq=1 ttl=64 time=0.585 ms

--- 10.0.4.3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.585/0.585/0.585/0.000 ms
[root@localhost ~]# ping -c 1 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=55 time=14.1 ms

--- 1.1.1.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 14.123/14.123/14.123/0.000 ms
[root@localhost ~]#
```

Figura 10: Pings a tots els dispositius de la xarxa

2 Comandes de VBoxManage per a la configuració

2.1 Assignar una tarja USB de les existents

Se'ns demana *mapejar* una tarjeta USB a la màquina virtual. Tot i que l'enunciat recomanava una targeta USB WiFi, en el meu cas no disposava d'aquest component, i per tant ho he fet amb un perifèric bluetooth (un ratolí).

Primerament, s'imprimeix la llista de màquines virtuals disponibles:

```
[pablo@archcrypted ~] VBoxManage list vms
"centos-firewall" {fe1267ba-af2e-43d6-912b-8c1f509e273b}
"centos-ids" {68b5705f-8d29-4f79-84fe-8876d5662f2e}
"centos-server-1" {6d42c255-eefe-49ab-92ad-16a32338b2fe}
"centos-server-2" {57df1891-2ecd-4842-8f52-816b11f05641}
```

A continuació, llistem les targetes usb amb la comanda:

```
[pablo@archcrypted ~] VBoxManage list usbhost
Host USB Devices:
......
UUID:               c932afd5-5873-46b5-9635-0359c420d210
VendorId:           0x1130 (1130)
ProductId:          0x1704 (1704)
Revision:           17.16 (1716)
Port:               1
USB version/speed:  2/Full
Manufacturer:       Tenx Technology, Inc.
Product:            2.4G Wireless Optical Mouse
Address:            sysfs:/sys/devices/pci0000:00/0000:00:14.0/usb1/
                    1-2//device:/dev/vboxusb/001/012
Current State:      Busy
.......
```

A continuació, activem els controladors usb 2.0 i 3.0 (aquest pas s'ha de fer amb la VM que volem habilitar els usb's apagada):

```
$ VBoxManage modifyvm <id-vm> --usb on --usbehci on --usbxhci on
```

```
[pablo@archcrypted ~] VBoxManage modifyvm fe1267ba-af2e-43d6-912b-8c1f509e273b
--usb on --usbehci on --usbxhci on
```

Finalment, afegim el dispositiu usb a la màquina:

```
$ VBoxManage controlvm <id-machine> usbattach <id-usb>
```

```
[pablo@archcrypted ~] VBoxManage controlvm fe1267ba-af2e-43d6-912b-8c1f509e273b
usbattach c932afd5-5873-46b5-9635-0359c420d210
```

Tal i com es veu a la figura 11, abans de la comanda anterior no es detectava el dispositiu usb, però després d'aplicar la comanda apareix el ratolí com a dispositiu usb.

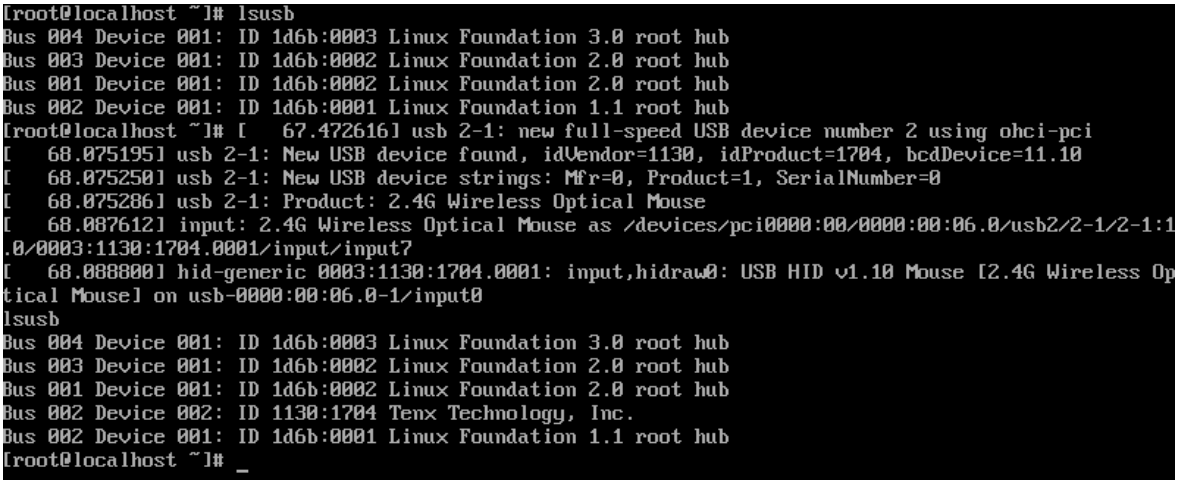

Figura 11: Reconeixement del ratolí a la VM

2.2 Mapejar un disc físic

S'identifica el disc amb la comanda:

```
$ lsblk
```

A continuació, es crea un disc virtual (Virtual Machine DisK) a partir del disc físic:

```
$ VBoxManage internalcommands createrawvmdk -filename <name>.vmdk -rawdisk /dev/<identifier>
```

A continuació, identifiquem la màquina virtual on volem mapejar el disc físic amb la ja coneguda:

```
$ VBoxManage list vms
```

I finalment, mapejem el disc:

```
$ VBoxManage storeattach --uuid <id VM>
--storagectl "IDE" --port 0 --device 0
--type hdd --medium <name>.vmdk
```

3 Clonatge de màquines

En aquesta secció s'explicaran tres possibles solucions al clonatge de màquines.

3.1 Clonezilla

Per tal de fer el clonatge de màquines, l'opció més recomanada i que acostuma a funcinar és emprar l'eina **Clonezilla**.

El procediment per a clonar les màquines seria el següent:

1. Es copia l'iso de clonezilla en un usb.
2. A la bios/uefi es configura el usb com el disc d'arrencada.
3. Un cop s'arrenca amb clonezilla, s'ha de seguir els passos de la GUI per a clonar el disc. Al finalitzar hauria de mostrar al disc on hem dit que es volia guardar el backup una carpeta amb *<any>-<mes>-<dia>-img*.
4. Un cop tinguem el backup, creem una màquina virtual on la ISO d'arrancada també es clonezilla.
5. Muntem a virtualbox el disc on tinguem la carpeta de backup.
6. Mitjançant clonezilla restaurem el backup a la màquina virtual.

Recalcar que en aquest cas és recomanable tenir una configuració de disc a la màquina virtual **dinàmica**.
Per tal de veure un procediment complet, es pot consultar aquest post.

3.2 Virtualbox

Amb la comanda VBoxManage tambe se'ns permet convertir un disc a un arxiu .vdi. El procediment és el següent:

```
$ cat /dev/<disk> |
VBoxManage convertfromraw stdin <VM-name>.vdi <size in bytes>
```

3.3 Comanda dd

Com a linux tot són fitxers, podem emprar la comanda *dd* (que serveix per a convertir i copiar fitxers) per tal de crear una copia del sistema operatiu i restablir-la a la màquina virtual.
El procediment es troba explicat en aquesta pàgina web.

4 Docker i Docker-Compose

Docker és un projecte de codi obert que permet crear contenidors de programari que s'executen de manera aïllada. Un contenidor és un paquet de programari que agrupa tot el codi, configuració i dependències d'una aplicació i que empra una part del sistema operatiu (concretament una part del kernel) de forma aïllada. D'aquesta manera, es poden desplegar aplicacions sense donar importància a les dependències que tingui instal·lat el host, el nostre programari serà capaç de funcionar en tota màquina que tingui docker instal·lat.

A nivell de seguretat, ens aporta una capa d'abstracció més que els atacants haurien de travessar per tal d'arribar fins a la màquina host [1].

A més, com els contenidors es troben aïllats entre ells, si el contenidor que executava el servidor es veu compromés, aquest no afectarà a les dades del contenidor de la base de dades, ja que es troben en namespaces diferents.

Docker-Compose és una eina que facilita als sys-admins/programadors crear i configurar contenidors de docker. S'acostuma a emprar per a crear xarxes entre contenidors, establir quins contenidors s'han d'estar executant abans que un altre, etc.

Com a conclusió, es pot dir que docker ens facilita poder ficar en producció un software, ja que configurem el contenidor a partir de Dockerfiles que sempre portaràn al mateix resultat de paquets instal·lats, dependències, etc. A més ens aporta seguretat ja que les diferents aplicacions es troben en namespaces separats (és a dir, no es veuen els processos entre sí tot i que comparteixin kernel).[2]

[1] Concretament, haurien de sortir del namespace que el kernel crea per aquell contenidor per a poder accedir als processos de la màquina host.

[2] Per això els contenidors són molt més ràpids que una màquina virtual. A la màquina virtual es necessita primer carregar el hypervisor, a continuació carregar el bootloader, el kernel, fins arribar a la capa d'aplicacions on s'executa el programa desitjat. En canvi un container únicament ha de crear el namespace i executar l'aplicació.