


|   |   |                           |   |
|---|---|---------------------------|---|
| Generalitat de Catalunya<br>Departament d'Educació<br><b>INS Provençana</b> | Mòdul 6:<br>Seguretat informàtica       | Curs<br>2023-2024         |  |
| Departament d'informàtica<br>Grup SMX2C                                     | UF5 - Tallafocs i monitoratge de xarxes | Nota                      |   |
|   | <b>NF1 - Monitoratge de xarxes</b>      | Data: 30 de Gener de 2024 |   |

**Professors:** Laura Montesinos

**Alumne:** Pol Juncà Lorente

## Instruccions

### Lliurament:

1. Poseu el **nom i cognoms** dins del document que lliureu.
2. Heu de lliurar un **únic document** en format **pdf**.
3. Heu de fer captures de pantalla per demostrar que heu realitzar l'exercici. Les captures de pantalla han d'anar annexades en aquest document.

**Consell:** Cada apartat té exercicis de diferent dificultat (feu primer els que sapigueu).

4. Durada: **2** hores.

- Pots consultar tot el material que consideris adient (també Internet)
- Tots els exercicis valen el mateix.
- Cada apartat té exercicis de diferent dificultat (fes primer els fàcils)

Crea una màquina virtual amb el teu nom (enlloc de Alumne) i el codi **alumne-245**

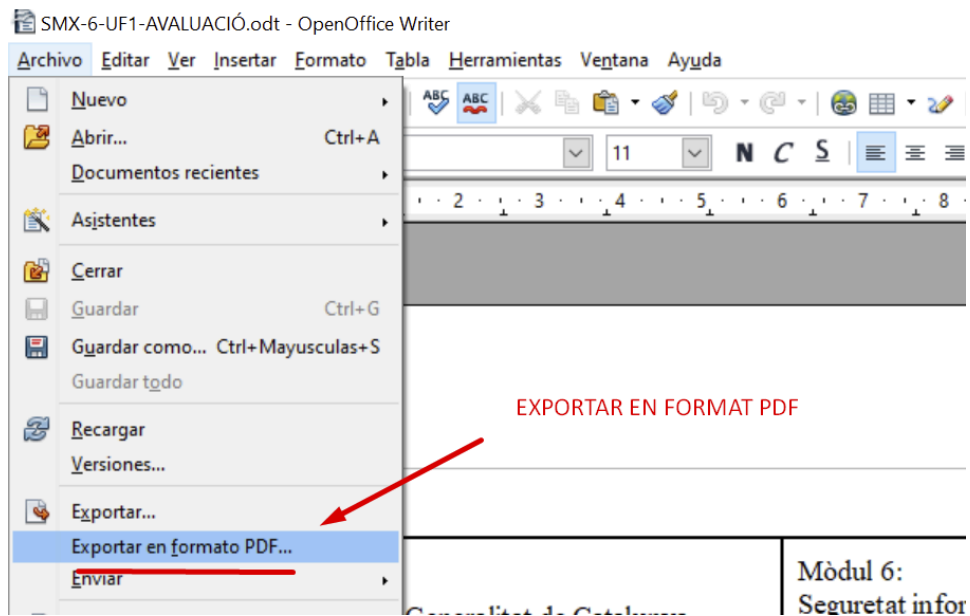
Arrenca la màquina configurant --cpu i --memory :

Inicieu la màquina.

Important. "Aneu al VirtualBox i poseu-hi adaptador pont".  
box.exe code laura-245 -c 4 -m 8000

|         |            |               |            |
|---------|------------|---------------|------------|
| C01-F25 | Versió 1.0 | Pàgina 1 de 2 | 10-09-2020 |
|---------|------------|---------------|------------|

El document l'has d'exportar a PDF:



## Exercicis

### 1. Docker-emmagatzematge- 25%

1.1. Borra tots els **contenidors** i verifica amb un `docker ps`.

```
Stop one or more running containers
box@pol-245:~$ docker rm $(docker ps -a -q)
6d6518f8c078
box@pol-245:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

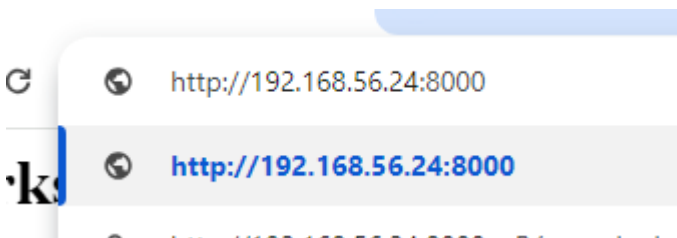
Borrem tots els contenidors amb a la comanda `docker rm $(docker ps -a -q)`

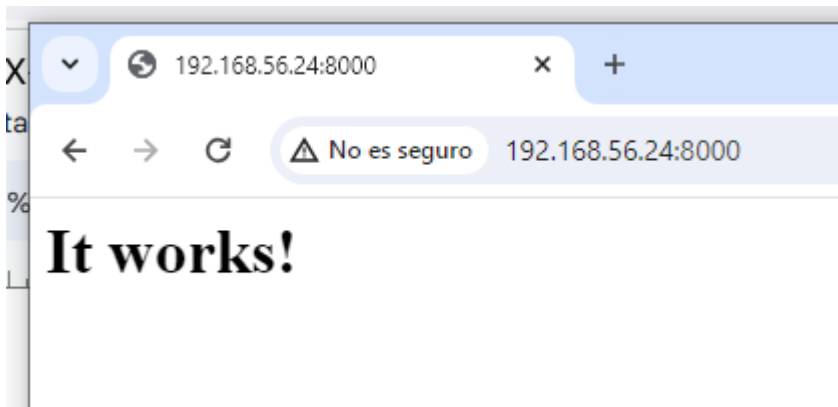
1.2. Arrenca un contenidor apache amb l'opció `-p 8000:80` perquè poguem connectar-nos des de fora del host. Verifica la pàgina d'inici del teu contenidor apache

```
7d91712d0c02
box@pol-245:~$ docker run -d -p 8000:80 --name apache_contenidor httpd
64f1620797c8b96f1260985400e047c5bd3c3172cb7e551fd088119635aa3d4e
box@pol-245:~$
```

Arrancem un docker apache amb el nom **apache\_contenidor**.

Obrim el navegador en la màquina local, posem la IP de l'interfície i el port 8000.





Podem veure que funciona.

- 1.3. Crea una carpeta amb el nom web i crea la pàgina d'inici amb el missatge "**La meva WEB: Nom\_alumne**"

```
box@pol-245:~$ mkdir web
box@pol-245:~$ echo "La meva WEB: Pol" > web/index.html
```

- 1.4. Torna a crear un nou contenidor **apache\_web** montant la carpeta **web**

```
box@pol-245:~$ docker run -d -p 8001:80 --name apache_web -v $(pwd)/web:/usr/local/apache2/htdocs
de35545a01b4331dbadf21c2fe3ff755e5833545e9a00a
box@pol-245:~$
```

Muntem la carpeta **web** a la carpeta on els servidors d'Apache es localitzen.

- 1.5. Fes un `curl` a localhost pots veure que ara respon amb el teu fitxer `index.html` no amb l'index d'inici, que has fet perquè això passi?

```
box@pol-245:~$ curl http://localhost:8001
La meva WEB: Pol
box@pol-245:~$ curl http://localhost:8000
<html><body><h1>It works!</h1></body></html>
box@pol-245:~$
```

Per a que ens surti l'índex que hem creat amb l'**echo**, hem de fer curl al port **8001** en comptes del port **8000**.

## 2. Docker-xarxes 25%

Per evitar problemes, primer elimina totes les xarxes que no s'estan fent servir:

```
docker network prune -f
```

2.1. Crea dues xarxes virtuals amb les següents ips:

net-1→10.0.10.0/24

net-2→ 10.0.20.0/24

```
box@pol-245:~$ docker network create --subnet=10.0.10.0/24 net-1
5de49f9d1bbad5c7126bb7bbae8ef3aeac5f8a70c610fa3a977148ef96308ed4
box@pol-245:~$ docker network create --subnet=10.0.20.0/24 net-2
28edf7a58e085480e6c5d014d5745350d6c5b305f6e2152684183a177d5ce399
box@pol-245:~$
```

Per crear la primera xarxa virtual., utilitzem la comanda **docker network create --subnet=10.0.10.0/24 net-1**

Per a la segona,

**docker network create --subnet=10.0.20.0/24 net-2**

2.2. Mostreu el resultat amb **docker network ls**.

```
box@pol-245:~$ docker network ls
NETWORK ID      NAME      DRIVER      SCOPE
29cae3126b09    bridge    bridge      local
2e364bb4acad    host      host        local
5de49f9d1bba    net-1     bridge      local
28edf7a58e08    net-2     bridge      local
87a97bed1177    none      null        local
box@pol-245:~$
```

Aquí podem veure les dues xarxes virtuals que acabem de crear.

### 2.3. Arrenca un contenidor amb nginx connectat a la primera xarxa-net-1

```
box@pol-245:~$ docker run -d --name nginx-container --network=net-1 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
2f44b7a888fa: Downloading  6.511MB/29.13MB
8b7dd3ed1dc3: Downloading  13.61MB/41.37MB
35497dd96569: Download complete
36664b6ce66b: Download complete
2d455521f76c: Download complete
dc9c4fdb83d6: Download complete
8056d2bcf3b6: Waiting
```

Per a arrancar un contenidor amb nginx, utilitzem la comanda **docker run -d --name nginx-container --network=net-1 nginx**.

Amb l'argument **--network=net-1** li especifiquem que assigni la xarxa net-1 al docker. Al introduir l'argument **nginx**, docker buscarà l'imatge de nginx localment, si no és trobada, la descarregarà d'internet.

Arrenca un contenidor amb apache connectat a la segona xarxa-net-2.

Fem el mateix procediment per a la xarxa **net-2** amb httpd(apache)

```
box@pol-245:~$ docker run -d --name apache-container --network=net-2 httpd
Unable to find image 'httpd:latest' locally
latest: Pulling from library/httpd
2f44b7a888fa: Already exists
376771e8483c: Pull complete
4f4fb700ef54: Pull complete
6a6627aecff0: Pull complete
152f4888b550: Pull complete
fd0579f22872: Pull complete
Digest: sha256:ba846154ade27292d216cce2d21f1c7e589f3b66a4a643bff0cdd348efd17aa3
Status: Downloaded newer image for httpd:latest
fa7792f288cdf63b0b3d13024ab889ef4ef78cbaddcbaea8ca9d552ddbd3d595
box@pol-245:~$
```

Comanda utilitzada: **docker run -d --name apache-container --network=net-2 httpd**

|         |            |               |            |
|---------|------------|---------------|------------|
| C01-F25 | Versió 1.0 | Pàgina 1 de 2 | 10-09-2020 |
|---------|------------|---------------|------------|

### 2.3.1. Verifica els contenidors

Amb la comanda **docker ps** comprovem que els dockers s'han creat correctament.

```
box@pol-245:~$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
fa7792f288cd   httpd      "httpd-foreground"      About a minute ago    Up About a minute    80/tcp       apache-container
a5b3e1f4e10f   nginx      "/docker-entrypoint..." 3 minutes ago        Up 3 minutes        80/tcp       nginx-container
box@pol-245:~$
```

### 2.4. Crear un contenidor amb nom **explorer** amb alpine

```
box@pol-245:~$ docker run -it --name explorer alpine
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
4abcf2066143: Pull complete
Digest: sha256:c5b1261d6d3e43071626931fc004f70149baeba2c8ec672bd4f27761f8e1ad6b
Status: Downloaded newer image for alpine:latest
box@pol-245:~$
```

Creem un contenidor sense assignar-lo a ninguna xarxa virtual, amb el nom **explorer** i l'imatge **alpine**.

Comanda utilitzada: **docker run -it --name explorer alpine**

#### 2.4.1. Connecta el contenidor explorer a les dues xarxes amb **docker network connect** i arrenca els contenidors:

```
box@pol-245:~$ docker network connect net-1 explorer
box@pol-245:~$ docker network connect net-2 explorer
box@pol-245:~$
```

Assignem les dues xarxes virtuals al docker **explorer** i engeguem els dockers.

```

box@pol-245:~$ docker start nginx-container
nginx-container
box@pol-245:~$ docker start apache-container
apache-container
box@pol-245:~$

```

## 2.5. Verifica les interfícies disponibles del docker

`docker exec explorer ip -f inet -4 -o add`

Per verificar les interfícies, utilitzem la comanda **docker exec explorer ip -f inet -4 -o addr**.

```

box@pol-245:~$ docker exec explorer ip -f inet -4 -o addr
1: lo      inet 127.0.0.1/8 scope host lo\          valid_lft forever preferred_lft forever
11: eth0    inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0\      valid_lft forever preferred_lft forever
13: eth1    inet 10.0.10.3/24 brd 10.0.10.255 scope global eth1\      valid_lft forever preferred_lft forever
15: eth2    inet 10.0.20.3/24 brd 10.0.20.255 scope global eth2\      valid_lft forever preferred_lft forever
box@pol-245:~$

```

## 3. Nmap 30%

- 3.1. Feu un escaneig del domini (escull un domini: [ibm.es](https://ibm.es), [movistar.es](https://movistar.es), [telefonica.es](https://telefonica.es)) i explica els resultats més rellevants:

```

box@pol-245:~$ nmap -sn ibm.es
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-30 18:55 UTC
Nmap scan report for ibm.es (199.46.34.76)
Host is up (0.12s latency).
Nmap done: 1 IP address (1 host up) scanned in 0.25 seconds
box@pol-245:~$

```

Per a fer un escaneig a **ibm.es**, utilitzarem la comanda **nmap** amb l'argument **sn**. Aquest argument és el que especifica fer un escaneig del host.



3.2. Mostra las ip's activas cercanas al domini XXXX/28 seleccionat abans.

Per veure les ip cercanes, podem utilitzar la comanda **dig** per a fer un escaneig:

```
box@pol-245:~$ dig ibm.es

; <<>> DiG 9.18.12-0ubuntu0.22.04.2-Ubuntu <<>> ibm.es
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43423
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 13, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;ibm.es.                                IN      A

;; ANSWER SECTION:
ibm.es.                                7111    IN      A      199.46.34.76

;; AUTHORITY SECTION:
                                7111    IN      NS      f.root-servers.net.
                                7111    IN      NS      k.root-servers.net.
                                7111    IN      NS      b.root-servers.net.
                                7111    IN      NS      m.root-servers.net.
                                7111    IN      NS      d.root-servers.net.
                                7111    IN      NS      g.root-servers.net.
                                7111    IN      NS      c.root-servers.net.
                                7111    IN      NS      a.root-servers.net.
                                7111    IN      NS      h.root-servers.net.
                                7111    IN      NS      e.root-servers.net.
                                7111    IN      NS      i.root-servers.net.
                                7111    IN      NS      l.root-servers.net.
                                7111    IN      NS      j.root-servers.net.

;; ADDITIONAL SECTION:
a.root-servers.net. 7111    IN      A      198.41.0.4

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Tue Jan 30 18:57:15 UTC 2024
;; MSG SIZE rcvd: 278

box@pol-245:~$
```

- 3.3. El domini **seleccionat** resuelve la IP XXXXXXXX? . Indica a quien pertenece el servidor detrás de la IP?

Per a veure a qui pertany el servidor, agafem la IP del servidor en la comanda anterior i executem un **whois**.

```
box@pol-245:~$ whois 198.41.0.4

#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/resources/registry/whois/tou/
#
# If you see inaccuracies in the results, please report at
# https://www.arin.net/resources/registry/whois/inaccuracy\_reporting/
#
# Copyright 1997-2024, American Registry for Internet Numbers, Ltd.
#

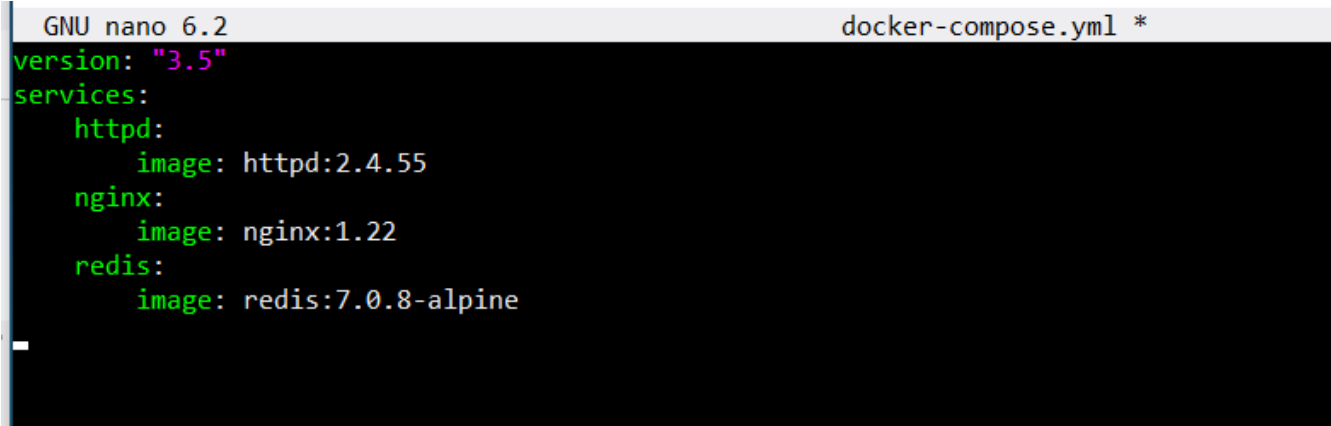
NetRange:      198.41.0.0 - 198.41.3.255
CIDR:          198.41.0.0/22
NetName:       INTERNIC1
NetHandle:     NET-198-41-0-0-1
Parent:        NET198 (NET-198-0-0-0-0)
NetType:       Direct Allocation
OriginAS:
Organization:  VeriSign Infrastructure & Operations (VIO-2)
RegDate:       1993-01-04
Updated:       2020-10-20
Ref:           https://rdap.arin.net/registry/ip/198.41.0.0

OrgName:       VeriSign Infrastructure & Operations
OrgId:         VIO-2
Address:       12061 Bluemont Way
City:          Reston
StateProv:     VA
PostalCode:    20190
Country:       US
RegDate:       2002-07-11
Updated:       2022-09-09
Ref:           https://rdap.arin.net/registry/entity/VIO-2
```

El servidor pertany a VeriSign.

3.4. - Crea aquest fitxer `docker-compose.yml`

```
version: "3.5"
services:
  httpd:
    image: httpd:2.4.55
  nginx:
    image: nginx:1.22
  redis:
    image: redis:7.0.8-alpine
```



The screenshot shows the Nano 6.2 text editor with the file `docker-compose.yml` open. The content of the file is the same as shown in the previous code block, with the text colorized in green on a black background. The editor's status bar at the top indicates 'GNU nano 6.2' and 'docker-compose.yml \*'.

```
GNU nano 6.2                                docker-compose.yml *
version: "3.5"
services:
  httpd:
    image: httpd:2.4.55
  nginx:
    image: nginx:1.22
  redis:
    image: redis:7.0.8-alpine
```

Creem l'arxiu amb Nano.

3.5. Executa la comanda `docker-compose up` en el mateix directori per aixecar el serveis definits en la configuració.

```

box@pol-245:~$ box@pol-245:~$ docker-compose up
+] Running 4/20
 0 nginx 6 layers [#####] 34.95MB/56.98MB Pulling
 0 f1f26f570256 Downloading [======>
 0 fd03b214f774 Downloading [=====
 0 ef2fc869b944 Download complete
 0 ac713a9ef2cc Download complete
 0 fd071922d543 Download complete
 0 2a9f38700bb5 Download complete
 0 redis 6 layers [#####] 0B/0B Pulling
 0 63b65145d645 Waiting
 0 6a83e1b979d3 Waiting
 0 33568fda55fd Waiting
 0 92c907937b14 Waiting
 0 ae96d2ab3885 Waiting
 0 fbd5435e8d0e Pulling fs layer
 0 httpd 5 layers [#####] 8.186MB/31.41MB Pulling
 0 3f9582a2cbe7 Downloading [======>
 0 9423d69c3be7 Waiting
 0 d1f584c02b5d Waiting
 0 8f73a485e312 Waiting
 0 b7697f8af320 Waiting

```

Executem el fitxer amb la comanda **docker-compose up**.

3.6. Mostra els dockers actius.

Comanda utilitzada: **docker ps**

```

Last login: Tue Jan 30 18:09:27 2024 from 192.168.56.1
box@pol-245:~$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
NAMES
7e62085c4e7a   redis:7.0.8-alpine                 "docker-entrypoint.s..." 38 seconds ago Up 33 seconds 6379/tcp
box-redis-1
f2a046f073ad   httpd:2.4.55                       "httpd-foreground"       38 seconds ago Up 33 seconds 80/tcp
box-httpd-1
5026a7b755ed   nginx:1.22                         "/docker-entrypoint...." 38 seconds ago Up 33 seconds 80/tcp
box-nginx-1
943ae8955569   httpd                              "httpd-foreground"       16 minutes ago Up 16 minutes 0.0.0.0:8001->80/tcp, :::8001->80/tcp
apache_web
64f1620797c8   httpd                              "httpd-foreground"       26 minutes ago Up 26 minutes 0.0.0.0:8000->80/tcp, :::8000->80/tcp
apache_contenedor
box@pol-245:~$

```

3.7. Obre un terminal interactiu en un contenidor i escaneja la xarxa privada

```
box@pol-245:~$ docker run -it --name terminal_interactiu ubuntu:latest
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
```

Creem el terminal interactiu i instal·lem nmap i iproute2

```
apt install -y nmap iproute2
```

```
root@9ca9761b33a3:/# apt install nmap iproute2
Reading package lists... 29%
```

Escanejem la nostra xarxa privada amb nmap.

```
root@9ca9761b33a3:/# nmap -p- -sV 192.168.56.24/24
Starting Nmap 7.80 ( https://nmap.org ) at 2024-01-30 19:16 UTC
Nmap scan report for 192.168.56.24
Host is up (0.000059s latency).
Not shown: 65532 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.3 (Ubuntu Linux; protocol 2.0)
8000/tcp  open  http     Apache httpd 2.4.58 ((Unix))
8001/tcp  open  http     Apache httpd 2.4.58 ((Unix))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.56.100
Host is up (0.00012s latency).
All 65535 scanned ports on 192.168.56.100 are filtered

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (2 hosts up) scanned in 54.57 seconds
root@9ca9761b33a3:/#
```

3.8. Explica la topografia de la xarxa privada/24 (ip, port , servei). Omple la següent taula.

| Host Name | Port   | IP            | Service | Version                 |
|-----------|--------|---------------|---------|-------------------------|
|           | 22/tcp | 192.168.56.24 | SSH     | OpenSSH 8.9p1<br>Ubuntu |

|  |          |               |      |   |
|--|----------|---------------|------|---|
|  |          |               |      | 3ubuntu0.3<br>(Ubuntu Linux;<br>protocol 2.0) |
|  | 8000/tcp | 192.168.56.24 | HTTP | Apache httpd<br>2.4.58 (Unix)                 |
|  | 8001/tcp | 192.168.56.24 | HTTP | Apache httpd<br>2.4.58 (Unix)                 |
|  |          |               |      |   |

### 3.9. Elimina tots els contenidors

```
box@pol-245:~$ docker rm -f $(docker ps -aq)
9ca9761b33a3
7e62085c4e7a
f2a046f073ad
5026a7b755ed
943ae8955569
64f1620797c8
box@pol-245:~$
```

Utilitzem la comanda **docker rm -f \$(docker ps -aq)** per eliminar tots els dockers.

## 4. TCPDUMP 20%

### 01 - tcpdump

#### 4.1. Mostra les interfícies que tens a la màquina.

```
box@pol-245:~$ tcpdump -D
1.enp0s3 [Up, Running, Connected]
2.enp0s8 [Up, Running, Connected]
3.any (Pseudo-device that captures on all interfaces) [Up, Running]
4.lo [Up, Running, Loopback]
5.bluetooth-monitor (Bluetooth Linux Monitor) [Wireless]
6.nflog (Linux netfilter log (NFLOG) interface) [none]
7.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
8.dbus-system (D-Bus system bus) [none]
9.dbus-session (D-Bus session bus) [none]
box@pol-245:~$
```

Per veure totes les interfícies, utilitzem la comanda **tcpdump -D**

#### 4.2. Obre un nou terminal i executa la comanda per veure tot el tràfic en una interfície

```
box@pol-245:~$ sudo tcpdump -i enp0s3
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
18:05:16.241079 Loopback, skipCount 0, invalid (256)
18:05:16.393983 IP pol-245.57711 > 192.168.124.5.domain: 65056+ [1au] A? motd.ubuntu.com. (44)
18:05:16.394217 IP pol-245.55193 > 192.168.124.5.domain: 7534+ [1au] AAAA? motd.ubuntu.com. (44)
18:05:16.406118 IP 192.168.124.5.domain > pol-245.57711: 65056 5/13/2 A 34.243.160.129, A 34.254.182.186, A 54.247.62.1, A 54.171.230.55, A 54.217.10.153 (351)
18:05:16.406118 IP 192.168.124.5.domain > pol-245.55193: 7534 5/13/2 AAAA 2a05:d018:91c:3200:c887:2f22:290f:a7c, AAAA 2a05:d018:91c:3200:c8f:1a06:a2dd:450f, AAAA 2a05:d018:91c:3200:5e0d:21a9:26ca:90b5, AAAA 2a05:d018:91c:3200:2846:99fb:81b6:1e11, AAAA 2a05:d018:91c:3200:d8b6:37bc:63f9:703c (411)
18:05:16.408130 IP pol-245.50194 > ec2-34-243-160-129.eu-west-1.compute.amazonaws.com.https: Flags [S], seq 2125799343, win 64240, options [mss 1460,sackOK,TS val 3286935958 ecr 0,nop,wscale 7], length 0
18:05:16.408144 IP pol-245.38161 > 192.168.124.5.domain: 41909+ [1au] PTR? 5.124.168.192.in-addr.arpa. (55)
18:05:16.416819 IP 192.168.124.5.domain > pol-245.38161: 41909 NXDomain* 0/1/1 (110)
18:05:16.416861 IP pol-245.38161 > 192.168.124.5.domain: 41909+ PTR? 5.124.168.192.in-addr.arpa. (44)
18:05:16.417590 IP 192.168.124.5.domain > pol-245.38161: 41909 NXDomain* 0/1/0 (99)
18:05:16.417980 IP pol-245.37003 > 192.168.124.5.domain: 24273+ [1au] PTR? 129.124.168.192.in-addr.arpa.
```

Per veure tot el tràfic, utilitzarem la comanda **tcpdump -i (interfície)**.

#### 4.3. Explica el resultat més importants del punt anterior

La comanda ens mostrarà informació detallada de cada paquet que passa per la interfície que hem seleccionat. També podem veure les IP d'origen i destí, el protocol utilitzat i la data de la sortida.

|         |            |               |            |
|---------|------------|---------------|------------|
| C01-F25 | Versió 1.0 | Pàgina 1 de 2 | 10-09-2020 |
|---------|------------|---------------|------------|

```

18:05:17.655120 IP pol-245.42396 > 192.168.124.5.domain: 27050+ PTR? 123.124.168.192.in-
18:05:17.656805 IP 192.168.124.5.domain > pol-245.42396: 27050 NXDomain* 0/1/0 (101)
18:05:17.657372 IP pol-245.47825 > 192.168.124.5.domain: 47120+ [1au] PTR? 6.124.168.192.
55)
18:05:17.658513 IP 192.168.124.5.domain > pol-245.47825: 47120 NXDomain* 0/1/1 (110)
18:05:17.658584 IP pol-245.47825 > 192.168.124.5.domain: 47120+ PTR? 6.124.168.192.in-ad
18:05:17.672367 IP 192.168.124.5.domain > pol-245.47825: 47120 NXDomain* 0/1/0 (99)
18:05:18.219153 IP 192.168.124.158.50806 > 239.255.255.250.1900: UDP, length 175
18:05:18.238811 Loopback, skipCount 0, invalid (256)
18:05:19.218932 IP 192.168.124.158.50806 > 239.255.255.250.1900: UDP, length 175
18:05:19.414802 IP6 fe80::a00:27ff:fe6c:c1bb > ip6-allrouters: ICMP6, router solicitation
18:05:20.214265 IP 192.168.124.158.50806 > 239.255.255.250.1900: UDP, length 175
18:05:20.239045 Loopback, skipCount 0, invalid (256)
18:05:20.770474 ARP, Request who-has pol-245 tell 192.168.124.5, length 46
18:05:20.770488 ARP, Reply pol-245 is-at 08:00:27:0c:de:de (oui Unknown), length 28
18:05:20.894327 ARP, Request who-has 192.168.124.5 tell pol-245, length 28
18:05:20.894868 ARP, Reply 192.168.124.5 is-at 7a:11:5f:dd:9b:57 (oui Unknown), length 40
18:05:21.662607 ARP, Request who-has _gateway tell pol-245, length 28
18:05:21.663149 ARP, Reply _gateway is-at 6a:80:d4:12:56:ac (oui Unknown), length 46
18:05:21.710773 IP pol-245.46911 > 192.168.124.5.domain: 42881+ [1au] PTR? 1.124.168.192.
55)

```

4.4. En un altre terminal executa un nslookup a google.es:

```

box@pol-245:~$ nslookup google.es
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   google.es
Address: 172.217.168.163
Name:   google.es
Address: 2a00:1450:4003:80c::2003

```

La comanda **nslookup** ens mostra informació sobre la resolució de noms del domini.

4.5. Executa la comanda que filtra el tràfic de totes les interfícies cuyo destí es dns excluyen el port ssh

El port utilitzat pel SSH és el 22. Per tant, especificarem en la comanda escoltar tots i exclur el port 22.



```
box@pol-245:~$ sudo tcpdump -i any not src port 22
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
18:13:31.662255 enp0s8 In IP 192.168.56.1.63866 > pol-245.ssh: Flags [.], ack 1664239477, v
gth 0
18:13:31.742043 lo In IP localhost.46187 > localhost.domain: 16611+ [1au] PTR? 24.56.16
.arpa. (55)
18:13:31.742106 enp0s8 Out IP pol-245.38475 > 192.168.134.5.domain: 51826+ [1au] PTR? 24.56
```

Comanda utilitzada: **tcpdump -i any not src port 22**