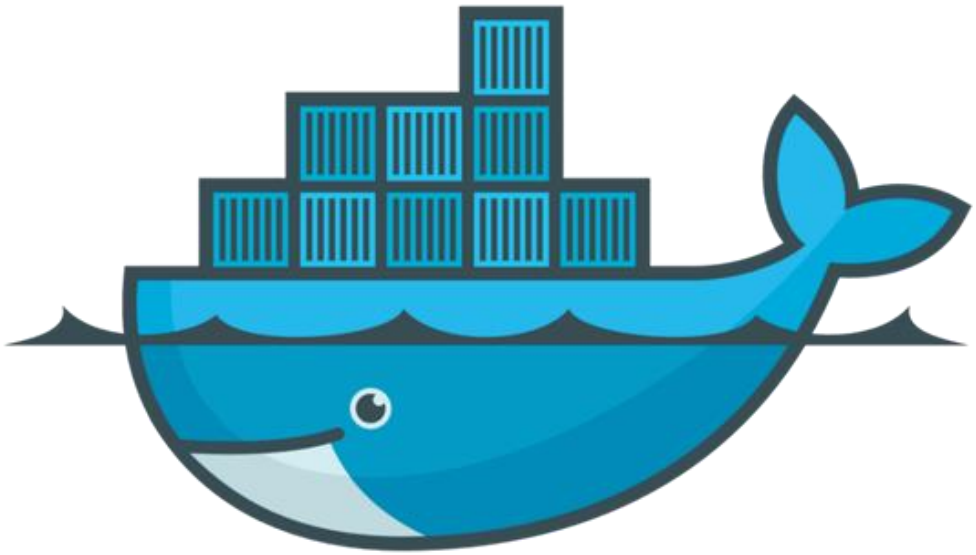
Tarea 1

DAW - 23/24



docker

➢ Docker. Imágenes

1) El tamaño que ocupa la imagen servidor web apache HTTP en su versión 2.4 bullseye en disco

a. 145MB.

```
windarlobogarces909@AdrianPC:~$ docker pull httpd:2.4-bullseye
2.4-bullseye: Pulling from library/httpd
759700526b78: Pull complete
e58f369f4099: Pull complete
9c6603403df2: Pull complete
a7b71700eed6: Pull complete
fb77a009c98c: Pull complete
Digest: sha256:4bdb49b01ab1a7b019410337588f772eebe6ae01d038987aa849fcb798fbf26f
Status: Downloaded newer image for httpd:2.4-bullseye
docker.io/library/httpd:2.4-bullseye

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview httpd:2.4-bullseye
windarlobogarces909@AdrianPC:~$ docker images httpd:2.4-bullseye
REPOSITORY   TAG            IMAGE ID       CREATED        SIZE
httpd        2.4-bullseye   6cd1f4feaa2f   8 months ago   145MB
```

2) ¿Cuál es el ID de la imagen que acabas de descargar?¿Y su hash o Digest? ¿Qué relación tienen entre sí?

```
windarlobogarces909@AdrianPC:~$  docker inspect 6cd1f4feaa2f
[
    {
        "Id": "sha256:6cd1f4feaa2f204ced0485dc5c641f4be1a5b1207804155514b219260ba7a82f",
        "RepoTags": [
            "httpd:2.4-bullseye"
        ],
        "RepoDigests": [
            "httpd@sha256:4bdb49b01ab1a7b019410337588f772eebe6ae01d038987aa849fcb798fbf26f"
        ],
        "Parent": "",
        "Comment": "",
        "Created": "2023-06-13T06:37:57.7462971257",
        "Container": "d1d60be03d34ae2840e7fdd4895505892c9b791b908bb8226fb3ed06f204602d",
        "ContainerConfig": {
            "Hostname": "d1d60be03d34",
            "Domainname": "",
            "User": "",
            "AttachStdin": false,
            "AttachStdout": false,
            "AttachStderr": false,
            "ExposedPorts": {
                "80/tcp": {}
            },
            "Tty": false,
            "OpenStdin": false,
            "StdinOnce": false,
            "Env": [
                "PATH=/usr/local/apache2/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
                "HTTPD_PREFIX=/usr/local/apache2",
                "HTTPD_VERSION=2.4.57",
                "HTTPD_SHA256=dbccb84aee95e095edfbb81e5eb926ccd24e6ada55dcd83caecb262e5cf94d2a",
                "HTTPD_PATCHES=rewrite-windows-testchar-h.patch 1d5620574fa03b483262dc5b9a66a6906553389952ab5d3070a02f887cc20193"
            ],
            "Cmd": [
                "/bin/sh",
                "-c",
                "#(nop) ",
                "CMD [\"httpd-foreground\"]"
            ],
            "Image": "sha256:ff473ee26bdd22eb6b72dcddf8d037f8e3e02f65e5f5bd34ada63c2445ffc733",
            "Volumes": null,
            "WorkingDir": "/usr/local/apache2",
            "Entrypoint": null,
            "OnBuild": null,
            "Labels": {},
            "StopSignal": "SIGWINCH"
        },
        "DockerVersion": "20.10.23",
        "Author": "",
        "Config": {
            "Hostname": "",
```

```
        "Entrypoint": null,
        "OnBuild": null,
        "Labels": {},
        "StopSignal": "SIGWINCH"
    },
    "DockerVersion": "20.10.23",
    "Author": "",
    "Config": {
        "Hostname": "",
        "Domainname": "",
        "User": "",
        "AttachStdin": false,
        "AttachStdout": false,
        "AttachStderr": false,
        "ExposedPorts": {
            "80/tcp": {}
        },
        "Tty": false,
        "OpenStdin": false,
        "StdinOnce": false,
        "Env": [
            "PATH=/usr/local/apache2/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
            "HTTPD_PREFIX=/usr/local/apache2",
            "HTTPD_VER=10m=2.4.57",
            "HTTPD_SHA256=dbccb84aee95e095edfbb81e5eb926ccd24e6ada55dcd83caecb262e5cf94d2a",
            "HTTPD_PATCHES=rewrite-windows-testchar-h.patch 1d562057f4a83b483262dc5b9a66a69865533899 52ab5d3070a02f887cc20193"
        ],
        "Cmd": [
            "httpd-foreground"
        ],
        "Image": "sha256:ff473ee26bdd22eb6b72dcddf8d037f8e3e82f65e5f5bd34ada63c2445ffc733",
        "Volumes": null,
        "WorkingDir": "/usr/local/apache2",
        "Entrypoint": null,
        "OnBuild": null,
        "Labels": null,
        "StopSignal": "SIGWINCH"
    },
    "Architecture": "amd64",
    "Os": "linux",
    "Size": 145205527,
    "GraphDriver": {
        "Data": {
            "LowerDir": "/var/lib/docker/overlay2/368ff79edda93497 35e9e3728b261b4db3db9cc310f6e3c6dc9e96bc26e074d9/diff:/var/lib/docker/overlay2/8b625c2ded74ac372be2d4a5fdf7ab1479da889b65ffa1268dc8f17e3f863ef7/diff:/var/lib/docker/overlay2/a54c86f2d85edc206d4a440ed89e6187 8d873b0be0eb542 6db74582184b279c/diff:/var/lib/docker/overlay2/3f3cde20b4fc3f72da9472e864b25922cead130c77e90eba32592b1246566c5e/diff",
            "MergedDir": "/var/lib/docker/overlay2/14c6910e29f82d6e9d53086c6204533b3ac889a1daeb23823770093ba635d5f5/merged",
            "UpperDir": "/var/lib/docker/overlay2/14c6910e29f82d6e9d53086c6204533b3ac889a1daeb23823770093ba635d5f5/diff",
            "WorkDir": "/var/lib/docker/overlay2/14c6910e29f82d6e9d53086c6204533b3ac889a1daeb23823770093ba635d5f5/work"
        },
        "Name": "overlay2"
    },
    "RootFS": {
        "Type": "layers",
        "Layers": [
            "sha256:0cc1f01656262cc1319655e8570146e4aa190c3fb8c7e81c353760c44a96c13b",
            "sha256:2b03e5e5516a3e89d034048eb8d9d8ee98efaaa88b3de4cd9e64924669d1f986",
            "sha256:e964f7a71e2c48fc9ab554c689332e1d6d4 3b4fbc2b89bf48374f388d0795f7414",
            "sha256:6618459c9c6bc5554205 01aaa8ab05d95627b6d22460b7abd8554173781beaed",
            "sha256:b8be365bdfa1d8cd1e60c54046aef62f37e5eca884 6fff9385861acd7f47399"
        ]
    },
    "Metadata": {
        "LastTagTime": "0001-01-01T00:00:00Z"
    }
  }
]
```


a. Digest:
httpd@sha256:4bdb49b01ab1a7b019410337588f772eebe6ae01d038987aa849fcb798fbf26f.
La cadena de hash es única de la imagen en el registro de Docker

b. El ID de la imagen:
feaa2f204ced0485dc5c641f4be1a5b1207804155514b219260ba7a82f.
El ID de la imagen es identificador único asignado por Docker

c. El digest se calcula basándose en el contenido único de la imagen y puede utilizarse para asegurar su integridad, lo que significa que cualquier cambio en el contenido generará un digest diferente. Por otro lado, el ID de la imagen es un identificador asignado internamente por Docker para su gestión operativa. Aunque son diferentes, el digest y el ID de la imagen están relacionados en el sentido de que ambos son importantes para el funcionamiento y la gestión adecuada de las imágenes Docker.

3) **Descarga ahora la versión 3.19 alpine de la imagen ¿Cuánto ocupa en disco? ¿Cuántas veces más pequeña o grande es la imagen de alpine respecto de la de bullseye? ¿A qué se debe esta diferencia?**

a. 7.38MB

```
windarlobogarces909@AdrianPC:~$ docker pull alpine:3.19
3.19: Pulling from library/alpine
4abcf2066143: Pull complete
Digest: sha256:c5b1261d6d3e43071626931fc004f70149baeba2c8ec672bd4f27761f8e1ad6b
Status: Downloaded newer image for alpine:3.19
docker.io/library/alpine:3.19

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview alpine:3.19
windarlobogarces909@AdrianPC:~$ docker images alpine:3.19
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
alpine       3.19      05455a08881e   5 weeks ago   7.38MB
```

b. La variación en el tamaño entre Alpine y Debian (Bullseye) se explica por varios factores. Esto incluye el diseño general de la distribución, la selección de paquetes, la distribución base y el gestor de paquetes utilizado en cada sistema. Alpine opta por musl libc y BusyBox, adoptando un enfoque minimalista en la selección de paquetes. Por otro lado, Debian (Bullseye) emplea glibc y ofrece un amplio conjunto de paquetes y herramientas para satisfacer una variedad de necesidades.

c. Alpine se concentra en ser eficiente y minimizar el uso de recursos. Por eso, suele ser más ligero y consumir menos recursos en comparación con Debian (Bullseye). Mientras tanto, Debian (Bullseye) está diseñado para ser versátil y ofrecer una amplia variedad de herramientas y paquetes para satisfacer diferentes necesidades.

4) **Elimina ambas imágenes del registro local**

```
windarlobogarces909@AdrianPC:~$ docker images
REPOSITORY   TAG             IMAGE ID       CREATED         SIZE
alpine       3.19            05455a08881e   5 weeks ago     7.38MB
httpd        2.4-bullseye    6cd1f4feaa2f   8 months ago    145MB
windarlobogarces909@AdrianPC:~$
```

```
windarlobogarces909@AdrianPC:~$ docker rmi httpd:2.4-bullseye
Untagged: httpd:2.4-bullseye
Untagged: httpd@sha256:4bdb49b01ab1a7b0194103375588f772eebe6ae01d038987aa849fcb798fbf26f
Deleted: sha256:6cd1f4feaa2f204ced0485dc5c641f4be1a5b1207804155514b219260ba7a82f
Deleted: sha256:a85c2fc24bfe2a595f7fae43655759a638c0f0dd21a39dbc1d33e78f888b0496
Deleted: sha256:c5cabd7c387a51fe36636b2c71d114c61a64ae7f53d95351bb947bd9d768c971
Deleted: sha256:19b474d2efbdf08e9cd23b775a9c23b3a1ed2adf36d02ccf57eff4fc435f8f48
Deleted: sha256:70cfc106cacab5c59075f610dc1ce1223b9df4aeb8f267970e1dc33eae434032
Deleted: sha256:0cc1f01656262cc1319655e8570146e4aa190c3fb8c7e81c353760c44a96c13b
windarlobogarces909@AdrianPC:~$ docker rmi alpine:3.19
Untagged: alpine:3.19
Untagged: alpine@sha256:c5b1261d6d3e43071626931fc004f70149baeba2c8ec672bd4f27761f8e1ad6b
Deleted: sha256:05455a08881ea9cf0e752bc48e61bbd71a34c029bb13df01e40e3e70e0d007bd
Deleted: sha256:d4fc045c9e3a848011de66f34b81f052d4f2c15a17bb196d637e526349601820
windarlobogarces909@AdrianPC:~$ docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
windarlobogarces909@AdrianPC:~$
```

➢ Docker. Containers y redes

1) Descarga la última versión de la imagen containous/whoami.

```
windarlobogarces909@AdrianPC: $  docker pull containous/whoami
Using default tag: latest
latest: Pulling from containous/whoami
29015087d73b: Pull complete
0109a80d13bc: Pull complete
d3cafff6f4d8: Pull complete
Digest: sha256:7d6a3c8f91470a23ef380320609ee6e69ac68d20bc804f3a1c6065fb56cfa34e
Status: Downloaded newer image for containous/whoami:latest
docker.io/containous/whoami:latest

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview containous/whoami
windarlobogarces909@AdrianPC: $ docker images
REPOSITORY          TAG       IMAGE ID       CREATED       SIZE
containous/whoami   latest    0f6fbbedd377   3 years ago   7.37MB
```

2) Levanta un contenedor llamado testserver1 a partir de la imagen anterior asignando un puerto aleatorio del sistema host (opción -P). Ejecútalo en segundo plano.

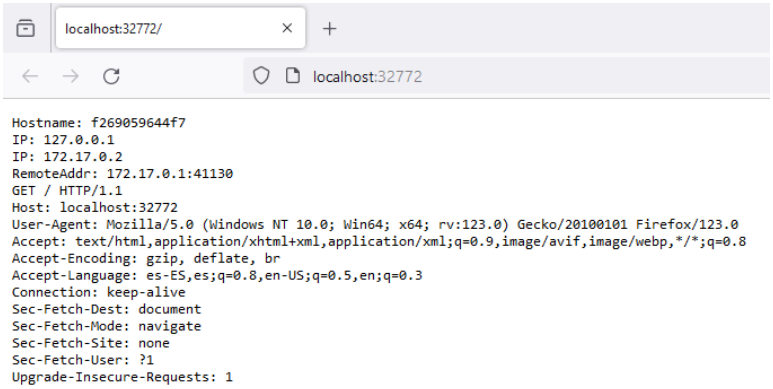
```
windarlobogarces909@AdrianPC: $  docker run -d -P --name testserver1 0f6fbbedd377
f269059644f77b25ac30fa4fcfc8e537654239b586ec4bd2886c5232789e487e
windarlobogarces909@AdrianPC: $ docker ps
CONTAINER ID   IMAGE          COMMAND     CREATED          STATUS          PORTS                   NAMES
f269059644f7   0f6fbbedd377   "/whoami"   16 seconds ago   Up 15 seconds   0.0.0.0:32772->80/tcp   testserver1
```

3) Inspecciona el contenedor anterior para determinar el puerto del host que se ha asignado. Accede a la URL http://localhost:PUERTO desde tu navegador

```
windarlobogarces909@AdrianPC: $  docker inspect f269059644f7
[
    {
        "Id": "f269059644f77b25ac30fa4fcfc8e537654239b586ec4bd2886c5232789e487e",
        "Created": "2024-03-08T19:40:54.673311905Z",
        "Path": "/whoami",
        "Args": [],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 9787,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2024-03-08T19:40:55.194997382Z",
            "FinishedAt": "0001-01-01T00:00:00Z"
        },
        "Image": "sha256:0f6fbbedd3777530ea3bedadf0a75b9aba805a55f6c5481ef0ebd762c5eeb818",
        "ResolvConfPath": "/var/lib/docker/containers/f269059644f77b25ac30fa4fcfc8e537654239b586ec4bd2886c5232789e487e/resolv.conf",
        "HostnamePath": "/var/lib/docker/containers/f269059644f77b25ac30fa4fcfc8e537654239b586ec4bd2886c5232789e487e/hostname",
        "HostsPath": "/var/lib/docker/containers/f269059644f77b25ac30fa4fcfc8e537654239b586ec4bd2886c5232789e487e/hosts",
        "LogPath": "/var/lib/docker/containers/f269059644f77b25ac30fa4fcfc8e537654239b586ec4bd2886c5232789e487e/f269059644f77b25ac30fa4fcfc8e537654239b586ec4bd2886c5232789e487e-json.log",
        "Name": "/testserver1",
        "RestartCount": 0,
```

```
        "RestartCount": 0,
        "Driver": "overlay2",
        "Platform": "linux",
        "MountLabel": "",
        "ProcessLabel": "",
        "AppArmorProfile": "",
        "ExecIDs": null,
        "HostConfig": {
            "Binds": null,
            "ContainerIDFile": "",
            "LogConfig": {
                "Type": "json-file",
                "Config": {}
            },
            "NetworkMode": "default",
            "PortBindings": {},
            "RestartPolicy": {
                "Name": "no",
                "MaximumRetryCount": 0
            },
            "AutoRemove": false,
            "VolumeDriver": "",
            "VolumesFrom": null,
            "ConsoleSize": [
                50,
                188
            ],
            "CapAdd": null,
            "CapDrop": null,
            "CgroupnsMode": "host",
            "Dns": [],
            "DnsOptions": [],
            "DnsSearch": [],
            "ExtraHosts": null,
            "GroupAdd": null,
            "IpcMode": "private",
            "Cgroup": "",
            "Links": null,
            "OomScoreAdj": 0,
            "PidMode": "",
            "Privileged": false,
            "PublishAllPorts": true,
            "ReadonlyRootfs": false,
            "SecurityOpt": null,
            "UTSMode": "",
            "UsernsMode": "",
            "ShmSize": 67108864,
            "Runtime": "runc",
            "Isolation": "",
            "CpuShares": 0,
```

```
                "Memory": 0,
                "NanoCpus": 0,
                "CgroupParent": "",
                "BlkioWeight": 0,
                "BlkioWeightDevice": [],
                "BlkioDeviceReadBps": [],
                "BlkioDeviceWriteBps": [],
                "BlkioDeviceReadIOps": [],
                "BlkioDeviceWriteIOps": [],
                "CpuPeriod": 0,
                "CpuQuota": 0,
                "CpuRealtimePeriod": 0,
                "CpuRealtimeRuntime": 0,
                "CpusetCpus": "",
                "CpusetMems": "",
                "Devices": [],
                "DeviceCgroupRules": null,
                "DeviceRequests": null,
                "MemoryReservation": 0,
                "MemorySwap": 0,
                "MemorySwappiness": null,
                "OomKillDisable": false,
                "PidsLimit": null,
                "Ulimits": [],
                "CpuCount": 0,
                "CpuPercent": 0,
                "IOMaximumIOps": 0,
                "IOMaximumBandwidth": 0,
                "MaskedPaths": [
                    "/proc/asound",
                    "/proc/acpi",
                    "/proc/kcore",
                    "/proc/keys",
                    "/proc/latency_stats",
                    "/proc/timer_list",
                    "/proc/timer_stats",
                    "/proc/sched_debug",
                    "/proc/scsi",
                    "/sys/firmware",
                    "/sys/devices/virtual/powercap"
                ],
                "ReadonlyPaths": [
                    "/proc/bus",
                    "/proc/fs",
                    "/proc/irq",
                    "/proc/sys",
                    "/proc/sysrq-trigger"
                ]
            },
            "GraphDriver": {
                "Data": {
                    "LowerDir": "/var/lib/docker/overlay2/29d636a8e01899c38afd11b0afb1e609a080dfb28f5e301a9a240317eb88bdef-init/diff:/var/lib/docker/overlay2/30e81bdc4879279f4c0e18e097ef2ae156
d4e3be2a5f3b23648ebef52a666323/diff:/var/lib/docker/overlay2/079db413abffb2a209b398c3d427dcf5538c91fecd4dca029db5f603ab1b2528/diff:/var/lib/docker/overlay2/9768298d84f111d52cbb6d49c9b34309
473b63ba63b3ce88317ddce07d799c8e/diff",
                    "MergedDir": "/var/lib/docker/overlay2/29d636a8e01899c38afd11b0afb1e609a080dfb28f5e301a9a240317eb88bdef/merged",
                    "UpperDir": "/var/lib/docker/overlay2/29d636a8e01899c38afd11b0afb1e609a080dfb28f5e301a9a240317eb88bdef/diff",
                    "WorkDir": "/var/lib/docker/overlay2/29d636a8e01899c38afd11b0afb1e609a080dfb28f5e301a9a240317eb88bdef/work"
                },
                "Name": "overlay2"
            },
            "Mounts": [],
            "Config": {
                "Hostname": "f269059644f7",
                "Domainname": "",
                "User": "",
                "AttachStdin": false,
                "AttachStdout": false,
                "AttachStderr": false,
                "ExposedPorts": {
                    "80/tcp": {}
                },
                "Tty": false,
                "OpenStdin": false,
                "StdinOnce": false,
                "Env": [
                    "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
                ],
                "Cmd": null,
                "Image": "0f6fbbedd377",
                "Volumes": null,
                "WorkingDir": "",
                "Entrypoint": [
                    "/whoami"
                ],
                "OnBuild": null,
                "Labels": {
                    "desktop.docker.io/wsl-distro": "Ubuntu"
                }
            },
            "NetworkSettings": {
                "Bridge": "",
                "SandboxID": "f7a1f5006748eb7f3b16f85ddbbc143f2f3bde33e3276fffd49a9fc471e4bf6a",
                "SandboxKey": "/var/run/docker/netns/f7a1f5006748",
                "Ports": {
                    "80/tcp": [
                        {
                            "HostIp": "0.0.0.0",
                            "HostPort": "32772"
                },
                "HairpinMode": false,
                "LinkLocalIPv6Address": "",
                "LinkLocalIPv6PrefixLen": 0,
                "SecondaryIPAddresses": null,
                "SecondaryIPv6Addresses": null,
                "EndpointID": "e560f36ab1c7e3d1976852dd9310ccd0f73bda8176b425818ee2e6d91476ed32",
                "Gateway": "172.17.0.1",
                "GlobalIPv6Address": "",
                "GlobalIPv6PrefixLen": 0,
                "IPAddress": "172.17.0.2",
                "IPPrefixLen": 16,
                "IPv6Gateway": "",
                "MacAddress": "02:42:ac:11:00:02",
                "Networks": {
                    "bridge": {
                        "IPAMConfig": null,
                        "Links": null,
                        "Aliases": null,
                        "MacAddress": "02:42:ac:11:00:02",
                        "NetworkID": "9085c687abbdf5524e88f323fc8ad5d69ff9a10ef615e8ae1eeaff948d6e3387",
                        "EndpointID": "e560f36ab1c7e3d1976852dd9310ccd0f73bda8176b425818ee2e6d91476ed32",
                        "Gateway": "172.17.0.1",
                        "IPAddress": "172.17.0.2",
                        "IPPrefixLen": 16,
                        "IPv6Gateway": "",
                        "GlobalIPv6Address": "",
                        "GlobalIPv6PrefixLen": 0,
                        "DriverOpts": null,
                        "DNSNames": null
                    }
                }
            }
        }
]
```



4) Crea un contenedor idéntico al anterior con el nombre testserver2, pero en esta ocasión enlaza los puertos 8080 y 4433 del host a los puertos 80 y 443 del contenedor .

a. Accede a las URLS http://localhost:8080 y https://localhost:4433 desde tu navegador

El puerto 443 esta en conflicto con el puerto 80, ambos puertos son estándar utilizados para el tráfico HTTP. Por lo general el puerto 80 se utiliza para HTTP sin cifrar, mientras que el puerto 443 se utiliza para HTTPS (Hyper Text Transfer Protocol Secure), que es HTTP seguro (van cifrados los datos). En si la imagen Whoami, solo funciona con conexiones HTTP.

5) Levanta un contenedor a partir de la imagen wbitt/network-multitool con el nombre nettools. No es necesario que publiques ningún puerto en el host

```
windarlobogarces909@AdrianPC:~$  docker run -d --name nettools wbitt/network-multitool
125d60077b676c1099c4c0ccfb78fdb93dcafe7e63289184a4fe1d72758ac93b
```

6) Utilizando el comando docker exec -it nettools bash accede al contenedor nettools y ejecuta el comando ping testserver1 ¿Obtienes respuesta? En caso negativo, ¿por qué no?

```
windarlobogarces909@AdrianPC:~$ docker exec -it nettools bash
125d60077b67:/#  ping testserver1
ping: testserver1: Name does not resolve
125d60077b67:/# exit
exit
```

a. No, no obtengo respuesta, al no estar en la misma red.
b. el contenedor testserver1 no tiene un servidor DNS configurado o no tiene una entrada de hosts que asocie el nombre "testserver1" con su dirección IP.

7) Repite el punto anterior pero en esta ocasión haz ping a la IP del contenedor testserver1 (tendrás que utilizar el comando docker inspect para conocer la IP) ¿Obtienes respuesta? ¿Por qué?

```
exit
windarlobogarces909@AdrianPC:~$  docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED          STATUS          PORTS                                                  NAMES
125d60077b67   wbitt/network-multitool   "/bin/sh /docker/ent…"   7 minutes ago    Up 7 minutes    80/tcp, 443/tcp, 1180/tcp, 11443/tcp                   nettools
1a767dd412e2   0f6fbbedd377             "/whoami"                20 minutes ago   Up 20 minutes   0.0.0.0:8080->80/tcp, 0.0.0.0:4433->443/tcp            testserver2
f269059644f7   0f6fbbedd377             "/whoami"                26 minutes ago   Up 26 minutes   0.0.0.0:32772->80/tcp                                  testserver1
windarlobogarces909@AdrianPC:~$  docker inspect  f269059644f7
[
    {
        "Id": "f269059644f77b25ac30fa4fcfc8e537654239b586ec4bd2886c5232789e487e",
        "Created": "2024-03-08T19:40:54.673311905Z",
        "Path": "/whoami",
        "Args": [],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 9787,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2024-03-08T19:40:55.194997382Z",
            "FinishedAt": "0001-01-01T00:00:00Z"
        },
        "Image": "sha256:0f6fbbedd3777530ea3bedadf0a75b9aba805a55f6c5481ef0ebd762c5eeb818",
        "ResolvConfPath": "/var/lib/docker/containers/f269059644f77b25ac30fa4fcfc8e537654239b586ec4bd2886c5232789e487e/resolv.conf",
        "HostnamePath": "/var/lib/docker/containers/f269059644f77b25ac30fa4fcfc8e537654239b586ec4bd2886c5232789e487e/hostname",
        "HostsPath": "/var/lib/docker/containers/f269059644f77b25ac30fa4fcfc8e537654239b586ec4bd2886c5232789e487e/hosts",
        "LogPath": "/var/lib/docker/containers/f269059644f77b25ac30fa4fcfc8e537654239b586ec4bd2886c5232789e487e/f269059644f77b25ac30fa4fcfc8e537654239b586ec4bd2886c5232789e487e-json.log",
        "Name": "/testserver1",
        "RestartCount": 0,
        "Driver": "overlay2",
        "Platform": "linux",
        "MountLabel": "",
        "ProcessLabel": "",
        "AppArmorProfile": "",
        "ExecIDs": null,
        "HostConfig": {
            "Binds": null,
```

```
        "MountLabel": "",
        "ProcessLabel": "",
        "AppArmorProfile": "",
        "ExecIDs": null,
        "HostConfig": {
            "Binds": null,
            "ContainerIDFile": "",
            "LogConfig": {
                "Type": "json-file",
                "Config": {}
            },
            "NetworkMode": "default",
            "PortBindings": {},
            "RestartPolicy": {
                "Name": "no",
                "MaximumRetryCount": 0
            },
            "AutoRemove": false,
            "VolumeDriver": "",
            "VolumesFrom": null,
            "ConsoleSize": [
                50,
                188
            ],
            "CapAdd": null,
            "CapDrop": null,
            "CgroupnsMode": "host",
            "Dns": [],
            "DnsOptions": [],
            "DnsSearch": [],
            "ExtraHosts": null,
            "GroupAdd": null,
            "IpcMode": "private",
            "Cgroup": "",
            "Links": null,
            "OomScoreAdj": 0,
            "PidMode": "",
            "Privileged": false,
            "PublishAllPorts": true,
            "ReadonlyRootfs": false,
            "SecurityOpt": null,
            "UTSMode": "",
            "UsernsMode": "",
            "ShmSize": 67108864,
            "Runtime": "runc",
            "Isolation": "",
            "CpuShares": 0,
            "Memory": 0,
            "NanoCpus": 0,
            "CgroupParent": "",
            "BlkioWeight": 0,
            "BlkioWeightDevice": [],
            "BlkioDeviceReadBps": [],
            "BlkioDeviceWriteBps": [],
            "BlkioDeviceReadIOps": [],
            "BlkioDeviceWriteIOps": [],
            "CpuPeriod": 0,
            "CpuQuota": 0,
            "CpuRealtimePeriod": 0,
            "CpuRealtimeRuntime": 0,
            "CpusetCpus": "",
            "CpusetMems": "",
            "Devices": [],
            "DeviceCgroupRules": null,
            "DeviceRequests": null,
            "MemoryReservation": 0,
            "MemorySwap": 0,
            "MemorySwappiness": null,
            "OomKillDisable": false,
            "PidsLimit": null,
            "Ulimits": [],
            "CpuCount": 0,
            "CpuPercent": 0,
            "IOMaximumIOps": 0,
            "IOMaximumBandwidth": 0,
            "MaskedPaths": [
                "/proc/asound",
                "/proc/acpi",
                "/proc/kcore",
                "/proc/keys",
                "/proc/latency_stats",
                "/proc/timer_list",
                "/proc/timer_stats",
                "/proc/sched_debug",
                "/proc/scsi",
                "/sys/firmware",
                "/sys/devices/virtual/powercap"
            ],
            "ReadonlyPaths": [
                "/proc/bus",
                "/proc/fs",
                "/proc/irq",
                "/proc/sys",
                "/proc/sysrq-trigger"
            ]
        },
        "GraphDriver": {
            "Data": {
                "LowerDir": "/var/lib/docker/overlay2/29d636a8e01899c38afd11b0afb1e609a080dfb28f5e301a9a240317eb88bdef-init/diff:/var/lib/docker/overlay2/30e81bdc4879279f4c0e18e897ef2ae156d4e3be2a5f3b23648ebef52a666323/diff:/var/lib/docker/overlay2/079db413abffb2a209b398c3d427dcf5538c91fecd4dca029db5f603ab1b2528/diff:/var/lib/docker/overlay2/9768298d84f111d52cbb6d49c9b34309473b63ba63b3ce88317ddce07d799c8e/diff",
                "MergedDir": "/var/lib/docker/overlay2/29d636a8e01899c38afd11b0afb1e609a080dfb28f5e301a9a240317eb88bdef/merged",
                "UpperDir": "/var/lib/docker/overlay2/29d636a8e01899c38afd11b0afb1e609a080dfb28f5e301a9a240317eb88bdef/diff",
                "WorkDir": "/var/lib/docker/overlay2/29d636a8e01899c38afd11b0afb1e609a080dfb28f5e301a9a240317eb88bdef/work"
            },
            "Name": "overlay2"
        },
        "Mounts": [],
        "Config": {
            "Hostname": "f26905644f7",
            "Domainname": "",
            "User": "",
            "AttachStdin": false,
            "AttachStdout": false,
            "AttachStderr": false,
            "ExposedPorts": {
                "80/tcp": {}
            },
            "Tty": false,
            "OpenStdin": false,
            "StdinOnce": false,
            "Env": [
                "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
            ],
            "Cmd": null,
            "Image": "0f6fbbedd377",
            "Volumes": null,
            "WorkingDir": "",
            "Entrypoint": [
                "/whoami"
            ],
            "OnBuild": null,
            "desktop.docker.io/wsl-distro": "Ubuntu"
        }
    },
    "NetworkSettings": {
        "Bridge": "",
        "SandboxID": "f7a1f5006748eb7f3b16f85ddbbc143f2f3bde33e3276fffd49a9fc471e4bf6a",
        "SandboxKey": "/var/run/docker/netns/f7a1f5006748",
        "Ports": {
            "80/tcp": [
                {
                    "HostIp": "0.0.0.0",
                    "HostPort": "32772"
                }
            ]
        },
        "HairpinMode": false,
        "LinkLocalIPv6Address": "",
        "LinkLocalIPv6PrefixLen": 0,
        "SecondaryIPAddresses": null,
        "SecondaryIPv6Addresses": null,
        "EndpointID": "e560f36ab1c7e3d1976852dd9310ccd0f73bda8176b425818ee2e6d91476ed32",
        "Gateway": "172.17.0.1",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
        "IPAddress": "172.17.0.2",
        "IPPrefixLen": 16,
        "IPv6Gateway": "",
        "MacAddress": "02:42:ac:11:00:02",
        "Networks": {
            "bridge": {
                "IPAMConfig": null,
                "Links": null,
                "Aliases": null,
                "MacAddress": "02:42:ac:11:00:02",
                "NetworkID": "9085c687abbdf5524e88f323fc8ad5d69ff9a10ef615e8ae1eeaff948d6e3387",
                "EndpointID": "e560f36ab1c7e3d1976852dd9310ccd0f73bda8176b425818ee2e6d91476ed32",
                "Gateway": "172.17.0.1",
                "IPAddress": "172.17.0.2",
                "IPPrefixLen": 16,
                "IPv6Gateway": "",
                "GlobalIPv6Address": "",
                "GlobalIPv6PrefixLen": 0,
                "DriverOpts": null,
                "DNSNames": null
            }
        }
    }
}
windarlobogarces909@AdrianPC: $
```

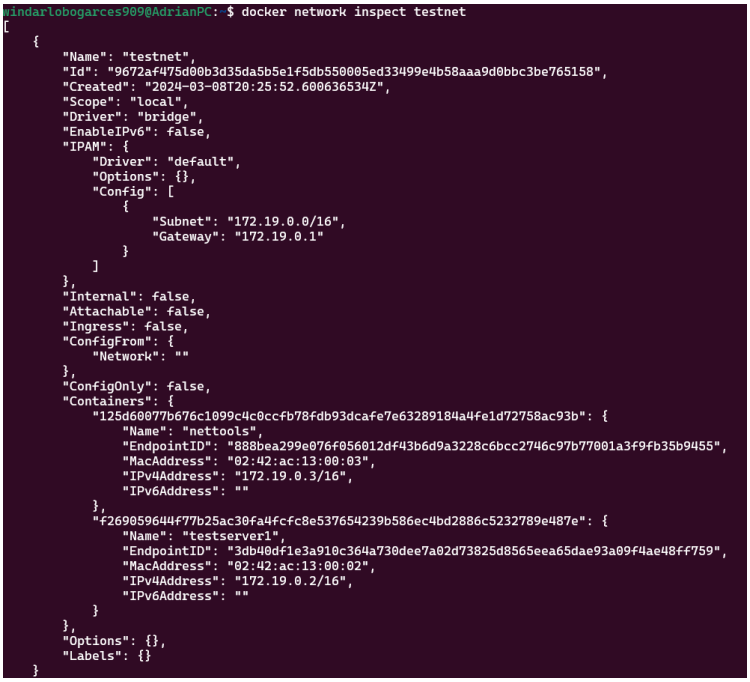
```
windarlobogarces909@AdrianPC:~$  docker exec -it nettools bash
125d60077b67:/#  ping 172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.125 ms
64 bytes from 172.17.0.2: icmp_seq=2 ttl=64 time=0.056 ms
64 bytes from 172.17.0.2: icmp_seq=3 ttl=64 time=0.051 ms
64 bytes from 172.17.0.2: icmp_seq=4 ttl=64 time=0.055 ms
64 bytes from 172.17.0.2: icmp_seq=5 ttl=64 time=0.079 ms
64 bytes from 172.17.0.2: icmp_seq=6 ttl=64 time=0.056 ms
64 bytes from 172.17.0.2: icmp_seq=7 ttl=64 time=0.051 ms
64 bytes from 172.17.0.2: icmp_seq=8 ttl=64 time=0.051 ms
64 bytes from 172.17.0.2: icmp_seq=9 ttl=64 time=0.056 ms
64 bytes from 172.17.0.2: icmp_seq=10 ttl=64 time=0.056 ms
64 bytes from 172.17.0.2: icmp_seq=11 ttl=64 time=0.059 ms
^[64 bytes from 172.17.0.2: icmp_seq=12 ttl=64 time=0.058 ms
64 bytes from 172.17.0.2: icmp_seq=13 ttl=64 time=0.070 ms
^X64 bytes from 172.17.0.2: icmp_seq=14 ttl=64 time=0.056 ms
z64 bytes from 172.17.0.2: icmp_seq=15 ttl=64 time=0.058 ms
64 bytes from 172.17.0.2: icmp_seq=16 ttl=64 time=0.056 ms
q64 bytes from 172.17.0.2: icmp_seq=17 ttl=64 time=0.065 ms
64 bytes from 172.17.0.2: icmp_seq=18 ttl=64 time=0.057 ms
64 bytes from 172.17.0.2: icmp_seq=19 ttl=64 time=0.058 ms
^Z
[1]+  Stopped                 ping 172.17.0.2
125d60077b67:/# exit
exit
There are stopped jobs.
125d60077b67:/# exit
exit
windarlobogarces909@AdrianPC:~$
```

a. Si obtengo respuesta ya que la comunicación entre los contenedores nettools y testserver1 se establece correctamente debido a que ambos contenedores están en la misma red de puente de Docker.

b. Porque cuando realizo el ping a la dirección IP del contenedor testserver1 desde el contenedor nettools, la comunicación se realiza a través de la red de puente de Docker, lo que permite que los contenedores se comuniquen entre sí.

8) Crea una red llamada testnet y une los contenedores testserver1 y nettools a la misma. Repite el apartado 6. Explica qué ocurre en esta ocasión

```
windarlobogarces909@AdrianPC:~$ docker network create testnet
9672af475d00b3d35da5b5e1f5db550005ed33499e4b58aaa9d0bbc3be765158
windarlobogarces909@AdrianPC:~$ docker network connect testnet testserver1
windarlobogarces909@AdrianPC:~$ docker network connect testnet nettools
windarlobogarces909@AdrianPC:~$  docker exec -it nettools bash
125d60077b67:/#  ping testserver1
PING testserver1 (172.19.0.2) 56(84) bytes of data.
64 bytes from testserver1.testnet (172.19.0.2): icmp_seq=1 ttl=64 time=0.081 ms
64 bytes from testserver1.testnet (172.19.0.2): icmp_seq=2 ttl=64 time=0.075 ms
64 bytes from testserver1.testnet (172.19.0.2): icmp_seq=3 ttl=64 time=0.072 ms
64 bytes from testserver1.testnet (172.19.0.2): icmp_seq=4 ttl=64 time=0.079 ms
64 bytes from testserver1.testnet (172.19.0.2): icmp_seq=5 ttl=64 time=0.085 ms
64 bytes from testserver1.testnet (172.19.0.2): icmp_seq=6 ttl=64 time=0.059 ms
64 bytes from testserver1.testnet (172.19.0.2): icmp_seq=7 ttl=64 time=0.070 ms
^C
--- testserver1 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6239ms
rtt min/avg/max/mdev = 0.059/0.074/0.085/0.007 ms
125d60077b67:/# exit
exit
```



a. Ocurre que Docker resuelve el nombre testserver1 a la dirección IP 172.19.0.2, que es la dirección IP asignada al contenedor testserver1 en la red testnet. Luego, el ping se realiza a esta dirección IP, lo que resulta en respuestas exitosas del contenedor testserver1. Al conectar los contenedores a la misma red, se crea un entorno de red privada donde los contenedores pueden comunicarse de manera eficiente y segura.

➢ Docker. Dockerfile y volúmenes

1) **Crea un Dockerfile que basado en la última versión de alpine de httpd que sirva una página web de muestra (no importa el contenido)**

```
windarlobogarces909@AdrianPC:~$ cd Tarea-1/
windarlobogarces909@AdrianPC:~/Tarea-1$  nano Dockerfile
```

```
  GNU nano 6.2
FROM httpd:alpine


COPY ./dist/ /usr/local/apache2/htdocs/
```

2) **Construye la imagen a partir del Dockerfile anterior con el tag {tunombre}/web5:latest**

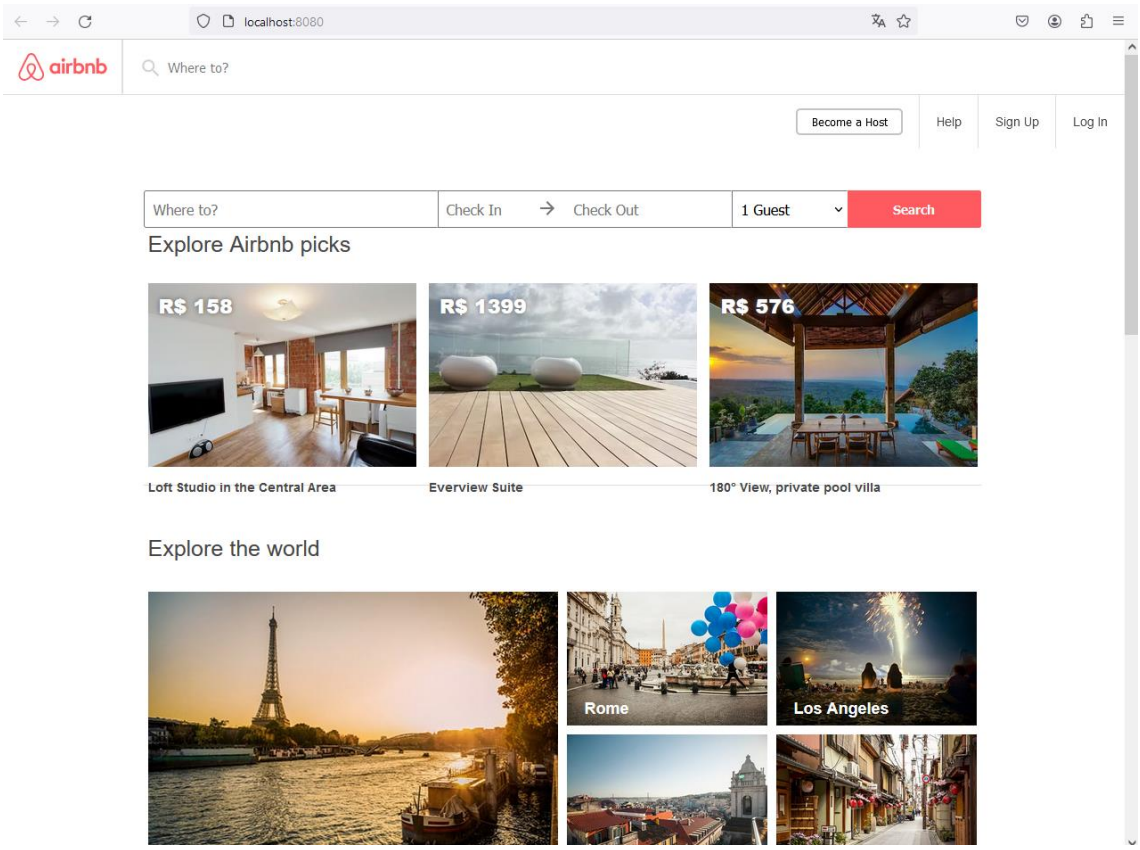
```
windarlobogarces909@AdrianPC:~/Tarea-1$  docker build -t windar/web5:latest .
[+] Building 0.6s (7/7) FINISHED                                                    docker:default
 => [internal] load build definition from Dockerfile                                          0.0s
 => => transferring dockerfile: 98B                                                           0.0s
 => [internal] load metadata for docker.io/library/httpd:alpine                               0.0s
 => [internal] load .dockerignore                                                             0.0s
 => => transferring context: 2B                                                               0.0s
 => [internal] load build context                                                             0.0s
 => => transferring context: 8.18kB                                                           0.0s
 => CACHED [1/2] FROM docker.io/library/httpd:alpine                                          0.0s
 => [2/2] COPY ./dist/ /usr/local/apache2/htdocs/                                             0.1s
 => exporting to image                                                                        0.1s
 => => exporting layers                                                                       0.1s
 => => writing image sha256:fa90bc2727146e4f5106da45a66d68f5baed74bac2c874b06f19ac0987dacbfa  0.0s
 => => naming to docker.io/windar/web5:latest                                                 0.0s

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview
```

3) **Levanta un contenedor de la imagen anterior llamado web5test que publique el puerto 80 en el 8080 del host. ¿Puedes acceder a la web enhttp://localhost:8080 ? En caso negativo ¿Qué está sucediendo? ¿Cómo lo arreglarías?**

```
windarlobogarces909@AdrianPC:~/Tarea-1$  docker run -d -p 8080:80  --name web5test  windar/web5
fee054f5be0ab1d44b5da4c56d8b4643ac55f23f9f8fe7226aca46c98ec5f322
windarlobogarces909@AdrianPC:~/Tarea-1$ docker ps
CONTAINER ID   IMAGE         COMMAND              CREATED         STATUS         PORTS                  NAMES
fee054f5be0a   windar/web5   "httpd-foreground"   4 seconds ago   Up 3 seconds   0.0.0.0:8080->80/tcp   web5test
windarlobogarces909@AdrianPC:~/Tarea-1$
```

a. Si puedo acceder a la pagina web



4) Detén el contenedor y elimínalo. Haz un cambio cualquiera en el fichero html de la web y vuelve a levantarlo (sin reconstruir la imagen). ¿Se aplicaron los cambios en la web? En caso negativo ¿qué ha sucedido?

```
windarlobogarces909@AdrianPC:~/Tarea-1$  docker stop web5test
web5test
windarlobogarces909@AdrianPC:~/Tarea-1$  docker rm -f web5test
web5test
windarlobogarces909@AdrianPC:~/Tarea-1$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

a. Realizo el cambio en el .search-btn

```
193  }
194  .search-btn {
195    float: left;
196    width: 150px;
197    height: 42px;
198    border: 1px solid #123ddb;
199    margin: 0 0 3px 0;
200    background-color: #3b15d6;
201    color: #fff;
202    font-weight: bold;
203    font-size: 14px;
204    border-top-right-radius: 3px;
205    border-bottom-right-radius: 3px;
206  }
```

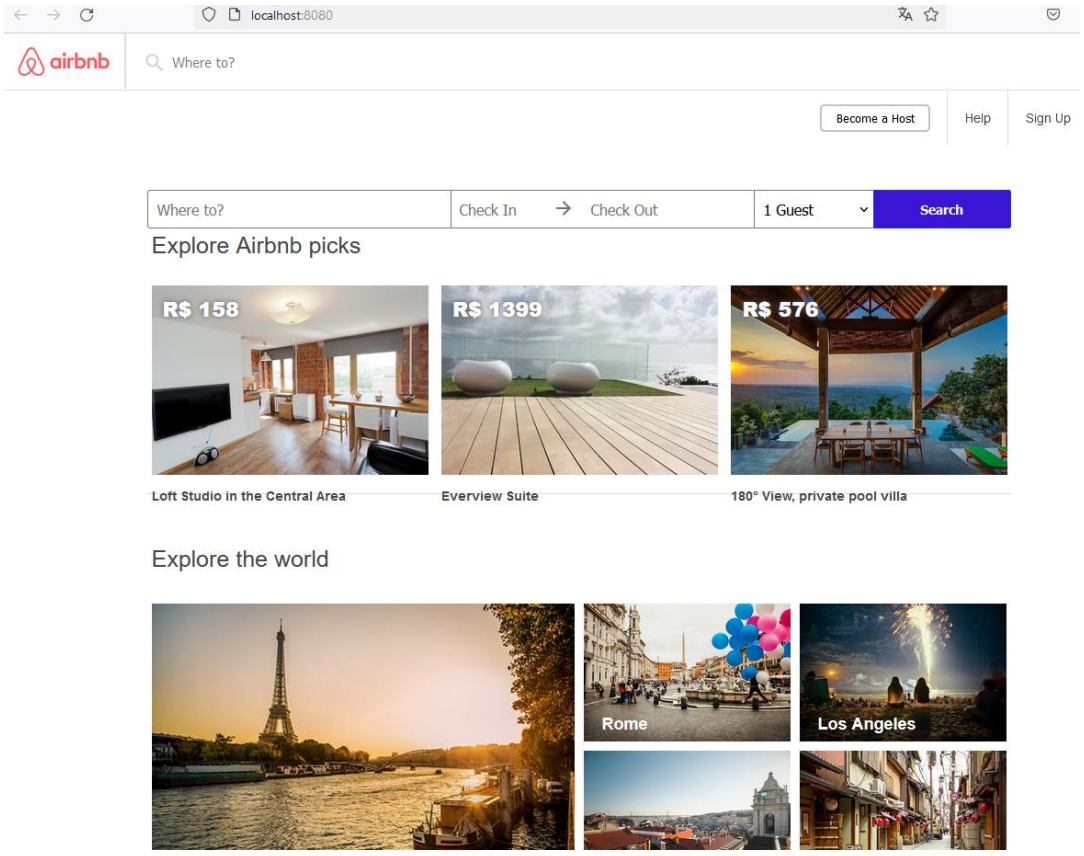
b. No se aplicaron los cambios, Docker utiliza una copia del sistema de archivos del contenedor al iniciar el contenedor descarta todos los cambios realizados dentro del contenedor cuando se detiene o elimina.

c. Sucede que si deseo mostrar los cambios es necesario utilizar volúmenes para montar directorios del host dentro del contenedor.

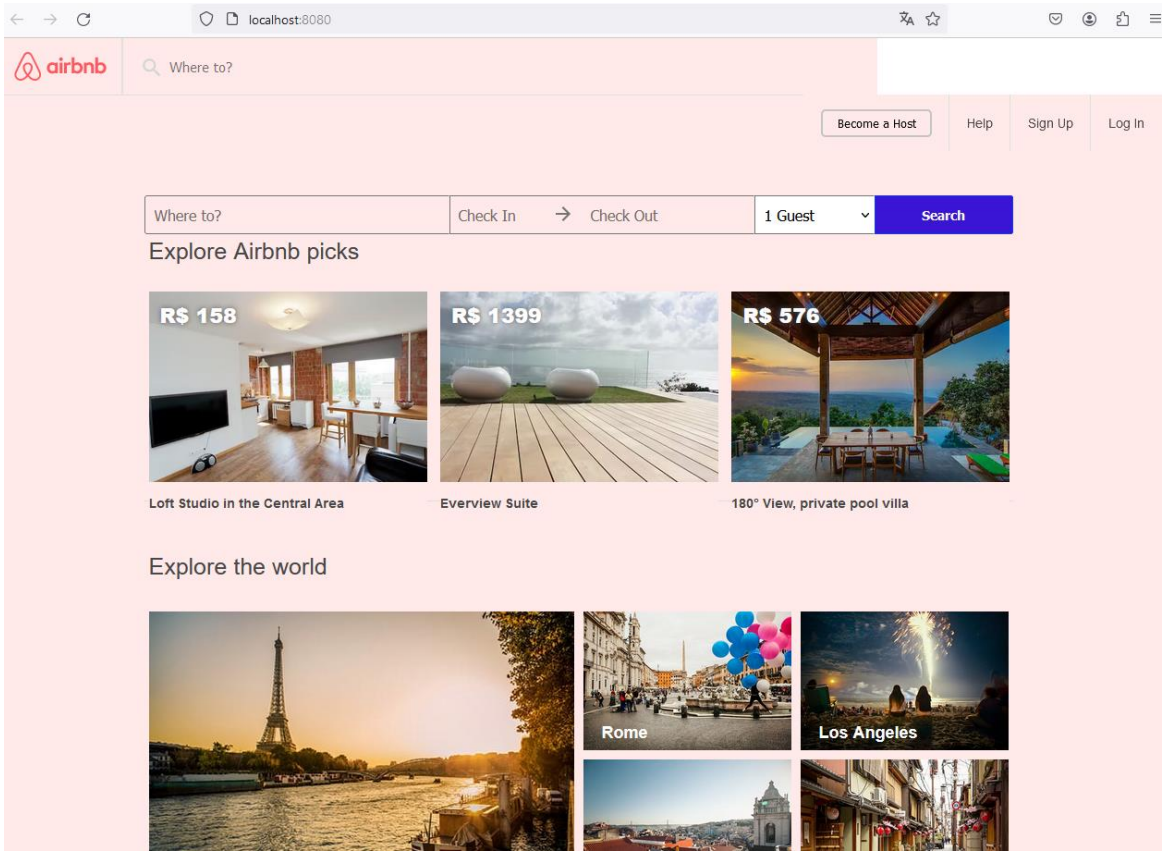
5) A continuación levanta un nuevo contenedor con el mismo nombre (web5test) pero en esta ocasión enlaza o mapea la ruta del host que contiene los estáticos de la web con el directorio del contenedor desde el que se sirven Este paso lo puedes realizar directamente por consola utilizando el comando docker run con la imagen oficial de httpd (es decir, puedes prescindir de la imagen custom construida a partir del Dockerfile). Comprueba que puedes acceder al sitio web que acabas de servir en localhost

```
windarlobogarces909@AdrianPC:~/Tarea-1$ docker rm -f web5test
web5test
windarlobogarces909@AdrianPC:~/Tarea-1$  docker run -d -p 8080:80  -v /home/windarlobogarces909/Tarea-1/dist:/usr/local/apache2/htdocs --name web5test windar/web5
fa74724377c9b767648f36988d43c2138f14feb1a9e2edcecab8c4956fe18139
windarlobogarces909@AdrianPC:~/Tarea-1$ docker ps
CONTAINER ID   IMAGE          COMMAND              CREATED          STATUS          PORTS                  NAMES
fa74724377c9   windar/web5    "httpd-foreground"   28 seconds ago   Up 27 seconds   0.0.0.0:8080->80/tcp   web5test
```

6) Sin eliminar o detener el contenedor anterior, modifica el fichero html y verifica que el cambio se aplica en el sitio web (puede ser necesario que actualices la página) ¿Se aplica el cambio?¿Por qué?



```
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  background-color: #ffeaea;
}
```



a. Ocurre que al colocar un volumen. Los cambios realizados en los archivos del host se reflejan dentro del contenedor y persisten incluso después de que el contenedor se detenga o se elimine y se conserva a lo largo del ciclo de vida del contenedor.