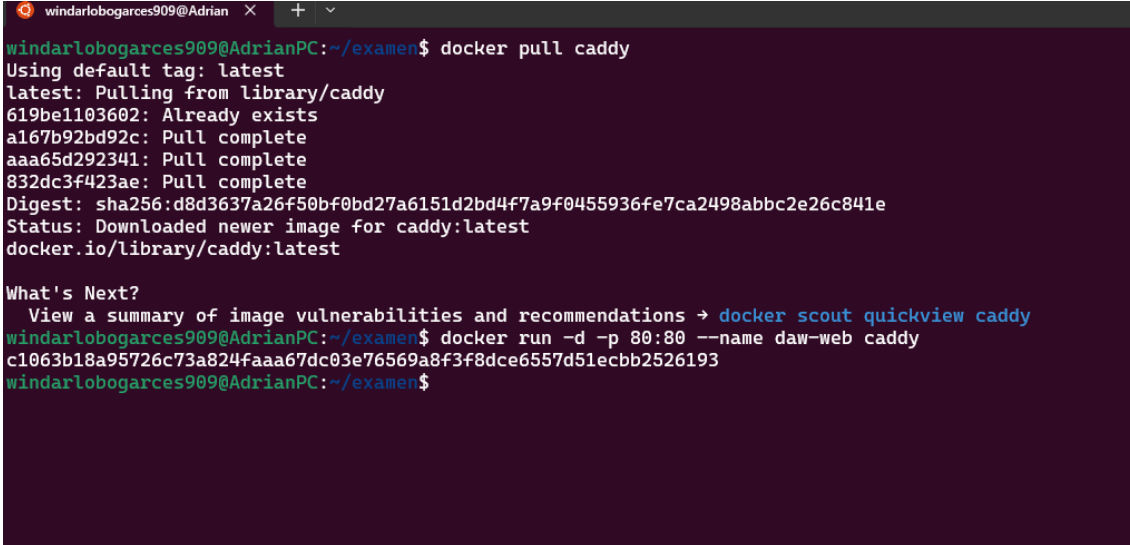


## 1. Docker (7 ptos)

1.1. Levanta un servidor web accesible por el puerto 80 del anfitrión basado en la imagen [https://hub.docker.com/\\_/caddy](https://hub.docker.com/_/caddy) (da igual el tag). El contenedor debe ejecutarse en modo “detached” y llamarse “daw-web”. No es necesario que sirva ninguna página web (0.75 ptos)



```
windarlobogarcas909@AdrianPC: ~/examen$ docker pull caddy
Using default tag: latest
latest: Pulling from library/caddy
619be1103602: Already exists
a167b92bd92c: Pull complete
aaa65d292341: Pull complete
832dc3f423ae: Pull complete
Digest: sha256:d8d3637a26f50bf0bd27a6151d2bd4f7a9f0455936fe7ca2498abbc2e26c841e
Status: Downloaded newer image for caddy:latest
docker.io/library/caddy:latest

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview caddy
windarlobogarcas909@AdrianPC: ~/examen$ docker run -d -p 80:80 --name daw-web caddy
c1063b18a95726c73a824faaa67dc03e76569a8f3f8dce6557d51ecbb2526193
windarlobogarcas909@AdrianPC: ~/examen$
```

1.2. ¿A qué red se encuentra conectado el contenedor anterior? ¿Cuál es la ip asignada al contenedor dentro de esta red? (0.75 ptos)

```

windartobogarc909@AdrianPC:~/exam$ docker inspect dave-web
[
  {
    "Id": "c1063b18a95726c73a824faaa67dc03e76569a8f3f8dce6557d51ecbb2526193",
    "Created": "2024-03-21T16:07:05.174510357Z",
    "Path": "caddy",
    "Args": [
      "run",
      "--config",
      "/etc/caddy/Caddyfile",
      "--adapter",
      "caddyfile"
    ],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 8824,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2024-03-21T16:07:05.65777764Z",
      "FinishedAt": "0001-01-01T00:00:00Z"
    },
    "Image": "sha256:0da4af4f9d73e970be3df873e00b8e658a7fc56ab982bffffa21d9bae68e3943",
    "ResolveConfPath": "/var/lib/docker/containers/c1063b18a95726c73a824faaa67dc03e76569a8f3f8dce6557d51ecbb2526193/resolve.conf",
    "HostnamePath": "/var/lib/docker/containers/c1063b18a95726c73a824faaa67dc03e76569a8f3f8dce6557d51ecbb2526193/hostname",
    "HostnamePath": "/var/lib/docker/containers/c1063b18a95726c73a824faaa67dc03e76569a8f3f8dce6557d51ecbb2526193/hostname",
    "HostsPath": "/var/lib/docker/containers/c1063b18a95726c73a824faaa67dc03e76569a8f3f8dce6557d51ecbb2526193/hosts",
    "LogPath": "/var/lib/docker/containers/c1063b18a95726c73a824faaa67dc03e76569a8f3f8dce6557d51ecbb2526193/c1063b18a95726c73a824faaa67dc03e76569a8f3f8dce6557d51ecbb2526193-json.log",
    "Name": "/dave-web",
    "RestartCount": 0,
    "Driver": "overlay2",
    "Platform": "linux",
    "MountLabel": "",
    "ProcessLabel": "",
    "AppArmorProfile": "",
    "ExecIDs": null,
    "HostConfig": {
      "Binds": null,
      "ContainerIDFile": "",
      "LogConfig": {
        "Type": "json-file",
        "Config": {}
      },
      "NetworkMode": "default",
      "PortBindings": {
        "80/tcp": [
          {
            "HostIp": "",
            "HostPort": "80"
          }
        ]
      },
      "RestartPolicy": {
        "Name": "no",
        "MaximumRetryCount": 0
      },
      "AutoRemove": false,
      "VolumeDriver": "",
      "VolumesFrom": null,
      "ConsoleSize": [
        30,
        120
      ],
      "CapAdd": null,
      "CapDrop": null,
      "CgroupsMode": "host",
      "Dns": [],
      "DnsOptions": [],
      "DnsSearch": [],
      "ExtraHosts": null,
      "GroupAdd": null,
      "IpcMode": "private",
      "Cgroup": "",
      "Links": null,
      "OomScoreAdj": 0,
      "PidMode": "",
      "Privileged": false,
      "PublishAllPorts": false,
      "ReadonlyRootfs": false,
      "SecurityOpt": null,
      "UTSMode": "",
      "UsersMode": "",
      "ShmSize": 67108864,
      "Runtime": "runc",
      "Isolation": "",
      "CpuShares": 0,
      "Memory": 0,
      "NanoCpus": 0
    }
  }
]

```

```

    "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/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/overlay2/47eb52e3086536fd15cde3f85c5f92f6f2a6b9e3c2bdee49cbde88fc79876abf-init/diff:/var/lib/docker/overlay2/b75d29eaf6067ac619ce5d2a5c8bd37c57083596064a9f18b1d475edd2fec7aa/diff:/var/lib/docker/overlay2/05b5d27e7b485efa81832c7dfbe02b1b4dd6e629cb2e0e1618d4926edac8e073/diff:/var/lib/docker/overlay2/ca28a1f5e1fa0628ee7b2499627f20a7ef0bf407f5d2b3bb84d94de6a6edc13b/diff:/var/lib/docker/overlay2/40a1831b8c911c6ffc6462d5c5576d793d53cd4aacc3f681f2435d0db7567e09/diff",
        "MergedDir": "/var/lib/docker/overlay2/47eb52e3086536fd15cde3f85c5f92f6f2a6b9e3c2bdee49cbde88fc79876abf/merged",
        "UpperDir": "/var/lib/docker/overlay2/47eb52e3086536fd15cde3f85c5f92f6f2a6b9e3c2bdee49cbde88fc79876abf/diff",
        "WorkDir": "/var/lib/docker/overlay2/47eb52e3086536fd15cde3f85c5f92f6f2a6b9e3c2bdee49cbde88fc79876abf/work"
    },
    "Name": "overlay2"
},
"Mounts": [],
"Config": {
    "Hostname": "c1063b18a957",
    "Domainname": "",
    "User": "",
    "AttachStdin": false,
    "AttachStdout": false,
    "AttachStderr": false,
    "ExposedPorts": {
        "2019/tcp": {}
    }
}

```

```

        "443/tcp": {},
        "443/udp": {},
        "80/tcp": {}
    },
    "Tty": false,
    "OpenStdin": false,
    "StdinOnce": false,
    "Env": [
        "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
        "CADDY_VERSION=v2.7.6",
        "XDG_CONFIG_HOME=/config",
        "XDG_DATA_HOME=/data"
    ],
    "Cmd": [
        "caddy",
        "run",
        "--config",
        "/etc/caddy/Caddyfile",
        "--adapter",
        "caddyfile"
    ],
    "Image": "caddy",
    "Volumes": null,
    "WorkingDir": "/srv",
    "Entrypoint": null,
    "OnBuild": null,
    "Labels": {
        "desktop.docker.io/wsl-distro": "Ubuntu",
        "org.opencontainers.image.description": "a powerful, enterprise-ready, open source web server with automatic HTTPS written in Go",
        "org.opencontainers.image.documentation": "https://caddyserver.com/docs",
        "org.opencontainers.image.licenses": "Apache-2.0",
        "org.opencontainers.image.source": "https://github.com/caddyserver/caddy-docker",
        "org.opencontainers.image.url": "https://caddyserver.com",
        "org.opencontainers.image.vendor": "Light Code Labs",
        "org.opencontainers.image.version": "v2.7.6"
    }
},
"NetworkSettings": {
    "Bridge": "",
    "SandboxID": "c0651d45d9db98fe4b0d2604c3658806f0b9bb51fcc19ade0d139265da5d0e2b",
    "SandboxKey": "/var/run/docker/netns/c0651d45d9db",
    "Ports": {
        "2019/tcp": null,
        "443/tcp": null,
        "443/udp": null,
        "80/tcp": [
            {
                "HostIp": "0.0.0.0",
                "HostPort": "80"
            }
        ]
    }
},
"HairpinMode": false,
"LinkLocalIPv6Address": "",
"LinkLocalIPv6PrefixLen": 0,
"SecondaryIPAddresses": null,
"SecondaryIPv6Addresses": null,
"EndpointID": "d5f913f87e284b9660b8a2d2d806017b0e07d40c29c11f307653f6ea160b9181",
"Gateway": "172.17.0.1",
"GlobalIPv6Address": "",
"GlobalIPv6PrefixLen": 0,
"IPAddress": "172.17.0.2",
"IPPrefixLen": 16,
"IPv6Gateway": "",
"MacAddress": "02:42:ac:11:00:02",
"Networks": {

```

1.3. Escribe un Dockerfile basado en la imagen del apartado 1.1 en el que copies los ficheros estáticos de una web cualquiera en el lugar adecuado para ser servidos (1 pto)

```
Dockerfile > FROM
1 FROM caddy
2
3 COPY ./dist/ /usr/share/caddy/
4
5
6
7
```

1.4. Construye una imagen con el tag “daw-web:v1.0.0”, y a partir de esta, ejecuta un contenedor que exponga por el puerto 8080 del anfitrión la web copiada en la etapa de construcción de la imagen. El contenedor debe llamarse “daw-web” (1 pto)

```
windarlobogarcas909@AdrianPC: ~/examen$ docker build -t daw-web:v1.0.0 .
[+] Building 0.3s (7/7) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 91B
=> [internal] load metadata for docker.io/library/caddy:latest
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build context
=> => transferring context: 123B
=> [1/2] FROM docker.io/library/caddy:latest
=> CACHED [2/2] COPY ./web /usr/share/caddy
=> exporting to image
=> => exporting layers
=> => writing image sha256:b4606e64c4718c4e1a8407198a093e223dbc09c8f3f77315d59e84b84d2e03bf
=> => naming to docker.io/library/daw-web:v1.0.0

Reduce build time with Docker Build Cloud: https://docs.docker.com/go/docker-build-cloud
windarlobogarcas909@AdrianPC: ~/examen$ docker run -d -p 8080:8080 --name daw-web daw-web:v1.0.0
14ca1bae3cf764efbd3dacda4138784e03cc4dc5c2bb62b2ff030684dcd5e3f8
windarlobogarcas909@AdrianPC: ~/examen$ docker run -d -p 8080:8080 --name daw-web daw-web:v1.0.0
8e6aa54ee4a50bd36211a7fa3119877860d8f965bd148aa254e10dale75dc579
windarlobogarcas909@AdrianPC: ~/examen$ docker run -d -p 8080:8080 --name daw-web daw-web:v1.0.0
1b80358483dd765579ca59933859f3e0dee255ef42a9dd66a00c716c6edc6604
windarlobogarcas909@AdrianPC: ~/examen$ docker run -d -p 8080:8080 --name daw-web daw-web:v1.0.0
32bc34e50187bf3ede5def4360e0f1667e0db98b6b3e1517c562186f732cdd48
windarlobogarcas909@AdrianPC: ~/examen$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
32bc34e50187   daw-web:v1.0.0   "caddy run --config ..."  2 minutes ago   Up 2 minutes   80/tcp, 443/tcp, 2019/tcp, 443/udp, 0.0.0.0:8080->8080/tcp
windarlobogarcas909@AdrianPC: ~/examen$ docker exec -it daw-web bash
```

1.5. Crea una red llamada “dawnnet” y conecta el contenedor del apartado 1.4 a la misma (0.5 ptos)

```
windarlobogarc909@AdrianPC:~/examen$ docker run -d -p 8080:80 --name daw-web daw-web:v1.0.0
607e5abc486731500e2fb769b4c6bf50d20c9ecf3fc9e2a4bcff311e1327e99b
windarlobogarc909@AdrianPC:~/examen$ docker network create dawnnet
41f2eaf53566d4985cb2fea226c3e515f113254047256e286167e2e731d6ab1f
windarlobogarc909@AdrianPC:~/examen$ docker network connect dawnnet daw-web
windarlobogarc909@AdrianPC:~/examen$

^Z*
windarlobogarc909@AdrianPC:~/examen$ docker run -d -it --name test-tool wbitt/network-multitool
Unable to find image 'wbitt/network-multitool:latest' locally
latest: Pulling from wbitt/network-multitool
7264a8db6415: Pull complete
5178053124a0: Pull complete
c9f6f349b303: Pull complete
e391ea244f6f: Pull complete
e5f6214c288d: Pull complete
Digest: sha256:d1137e87af76ee15cd0b3d4c7e2fcd111ffbd510ccd0af076fc98dddfc50a735
Status: Downloaded newer image for wbitt/network-multitool:latest
4badde72b3e28c952efb8d16e2a32b9a4189c06e28f9cc74c849e07fb2274a05
```

## 1.6. Ejecuta un contenedor basado en la imagen

<https://hub.docker.com/r/wbitt/network-multitool/tags> (da igual el tag) con el nombre

“test-tool” accede al contenedor con el comando “bash” y trata de hacer ping a la IP

del contenedor del apartado 1.4 (“daw-web”). ¿Qué resultado obtienes? ¿Qué está

ocurriendo? (1.5 ptos)

```
windarlobogarc909@AdrianPC:~/examen$ docker exec -it test-tool bash
4badde72b3e2:/# ping 192.168.128.2
PING 192.168.128.2 (192.168.128.2) 56(84) bytes of data.
█
```

No están en la misma red

## 1.7. Utilizando el comando “docker push” sube la imagen que construiste en el

apartado 1.4 a un repositorio público del registro por defecto Docker Hub (1 pto)

```
windarlobogarc909@AdrianPC:~/examen$ docker tag daw-web:v1.0.0 tu_usuario/daw-web:v1.0.0
windarlobogarc909@AdrianPC:~/examen$ docker push tu_usuario/daw-web:v1.0.0
The push refers to repository [docker.io/tu_usuario/daw-web]
dd716ad73b38: Preparing
c4cc1d8ff225: Preparing
e295d0b8eb53: Preparing
b000698907b5: Preparing
aedc3bda2944: Preparing
denied: requested access to the resource is denied
windarlobogarc909@AdrianPC:~/examen$ █
```

## 1.8. Elimina todos los contenedores, redes, volúmenes

```
windarlobogarcas909@AdrianPC: ~/examen$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
4badde72b3e2   wbit/network-multitool             "/bin/sh /docker/ent..." 19 minutes ago Up 19 minutes 80/tcp, 443/tcp, 1180/tcp, 11443/t
cp            test-tool
607e5abc4867   daw-web:v1.0.0                    "caddy run --config ..." 31 minutes ago Up 31 minutes 443/tcp, 2019/tcp, 443/udp, 0.0.0.
0:8080->80/tcp   daw-web
windarlobogarcas909@AdrianPC: ~/examen$ docker rm -f test-tool   daw-web
test-tool
daw-web
windarlobogarcas909@AdrianPC: ~/examen$ docker images
REPOSITORY      TAG       IMAGE ID       CREATED        SIZE
daw-web         v1.0.0    b4606e64c471   44 minutes ago 49.1MB
tu_usuario/daw-web v1.0.0    b4606e64c471   44 minutes ago 49.1MB
caddy           latest    0da4af4af9d7   5 days ago     49.1MB
wbit/network-multitool latest    713337546be6   6 months ago   75.9MB
windarlobogarcas909@AdrianPC: ~/examen$ docker rmi b4606e64c471 b4606e64c471 0da4af4af9d7 713337546be6
Untagged: caddy:latest
Untagged: caddy@sha256:d8d3637a26f50bf0bd27a6151d2bd4f7a9f0455936fe7ca2498abb2e26c841e
Deleted: sha256:0da4af4af9d73e970be3df873e00b8e658a7fc56ab982bffa21d9bae68e3943
Untagged: wbit/network-multitool:latest
Untagged: wbit/network-multitool@sha256:d1137e87af76ee15cd0b3d4c7e2fcd111fbd510ccd0af076fc98dddfc50a735
Deleted: sha256:713337546be623588ed8ff6d5e15dd3ccde8e4555ac5c97e5715d03580d2824
Deleted: sha256:104db9d3fb4a5a6eba0b69817d5f0a8d95300edb38960dc00265407ecc34a150
Deleted: sha256:00d62b3156baad61ccccfcbaf69f916d292e7e2d7f25505840d6bd36785872c
Deleted: sha256:97510635e1c199dc05e0ed199e565cac9fc2c76bdd494cea2ad1ffc2f97bddd35
Deleted: sha256:c9dc72d52166203b3d84a7ec5cc44cd31147bc8157335df3a805feb545f6e4e
Deleted: sha256:4693057ce2364720d39e57e85a5b8e0bd9ac3573716237736d6470ec5b7b7230
Error response from daemon: conflict: unable to delete b4606e64c471 (must be forced) - image is referenced in multiple repositories
Error response from daemon: conflict: unable to delete b4606e64c471 (must be forced) - image is referenced in multiple repositories
windarlobogarcas909@AdrianPC: ~/examen$
```

## 2. Docker Compose (3 ptos)

Escribe un fichero docker-compose.yaml con los siguientes servicios:

- 1 base de datos mysql (basado en la imagen oficial, no importa el tag)
- Accesible desde el puerto 3306 del anfitrión
- Que persista los datos en un volumen de docker llamado "dbdata"
- Que se cargue inicialmente desde base de datos de prueba worldb (<https://dev.mysql.com/doc/index-other.html>)
- 1 cliente de base de datos desde la imagen [https://hub.docker.com/\\_/phpmyadmin](https://hub.docker.com/_/phpmyadmin)
- Accesible desde el puerto 8080 del anfitrión

Todos lo

s servicios deben conectarse a una red llamada "composenet"

```

docker-compose.yml
1  version: "3.8"
2  services:
3    mysql:
4      image: mysql
5      volumes:
6        - dbdata:/var/lib/mysql
7        - ../world-db:/docker-entrypoint-initdb.d"
8      environment:
9        MYSQL_DATABASE: world
10       MYSQL_USER: root
11       MYSQL_PASSWORD: example
12       MYSQL_ROOT_PASSWORD: example
13      ports:
14        - "3306:3306"
15      networks:
16        - composenet
17
18     phpmyadmin:
19       image: phpmyadmin/phpmyadmin
20       environment:
21         PMA_HOST: mysql
22         PMA_USER: root
23         PMA_PASSWORD: example
24       ports:
25         - "8080:80"
26       depends_on:

```

```

❖ docker-compose.yml
2  services:
3    mysql:
15     networks:
17
18    phpmyadmin:
19     image: phpmyadmin/phpmyadmin
20     environment:
21       PMA_HOST: mysql
22       PMA_USER: root
23       PMA_PASSWORD: example
24     ports:
25       - "8080:80"
26     depends_on:
27       - mysql
28     networks:
29       - composenet
30
31   networks:
32     composenet:
33       driver: bridge
34
35   volumes:
36     dbdata:
37       driver: local
38

```