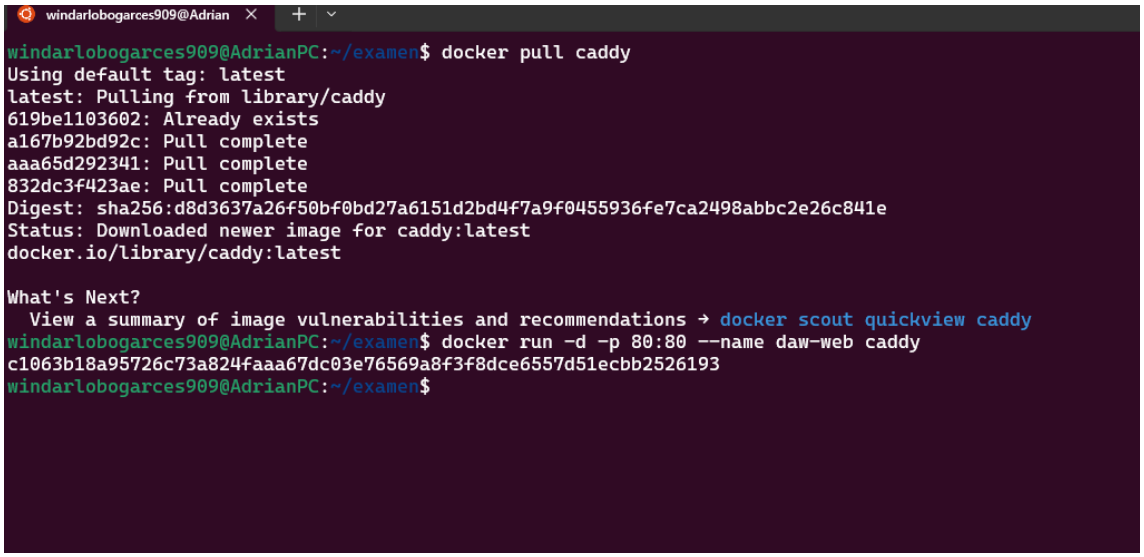
1. Docker (7 ptos)

1.1. Levanta un servidor web accesible por el puerto 80 del anfitrión basado en la imagen https://hub.docker.com/_/caddy (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)



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

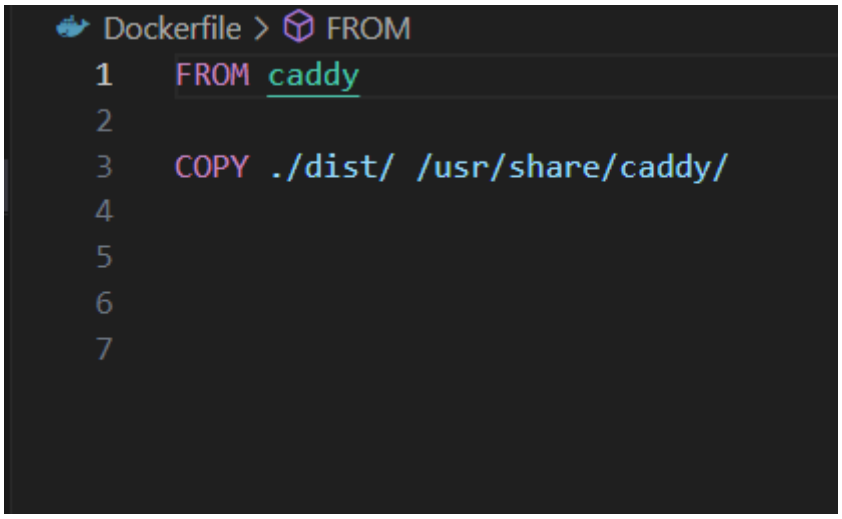
```
windarlobogarces909@AdrianPC:~/examen$ docker inspect daw-web
[
    {
        "Id": "c1063b18a95726c73a824faaa67dc03e76569a8f3f8dce6557d51ecbb2526193",
        "Created": "2024-03-21T16:07:05.174510357Z",
        "Path": "caddy",
        "Args": [
            "run",
            "--config",
            "/etc/caddy/Caddyfile",
            "--adapter",
            "caddyfile"
        ],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 8824,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2024-03-21T16:07:05.657777764Z",
            "FinishedAt": "0001-01-01T00:00:00Z"
        },
        "Image": "sha256:0da4af4af9d73e970be3df873e00b8e658a7fc56ab982bfffa21d9bae68e3943",
        "ResolvConfPath": "/var/lib/docker/containers/c1063b18a95726c73a824faaa67dc03e76569a8f3f8dce6557d51ecbb2526193/resolv.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/c1063b18a95726c73a8
7dc03e76569a8f3f8dce6557d51ecbb2526193-json.log",
    "Name": "/daw-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
```

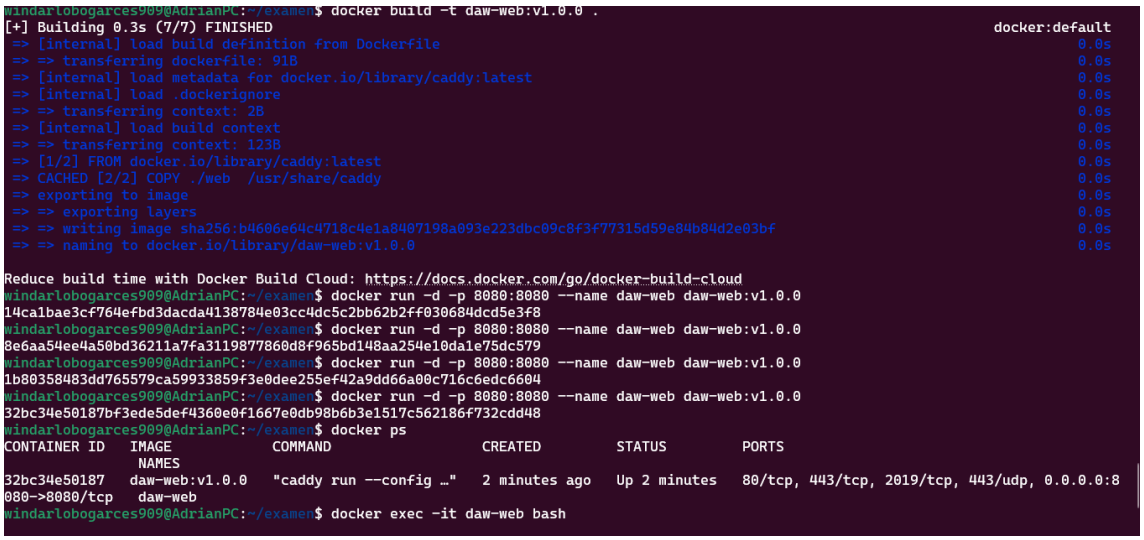
```
        },
        "AutoRemove": false,
        "VolumeDriver": "",
        "VolumesFrom": null,
        "ConsoleSize": [
            30,
            120
        ],
        "CapAdd": null,
        "CapDrop": null,
        "CgroupnsMode": "host",
        "Dns": [],
        "DnsOptions": [],
        "DnsSearch": [],
        "ExtraHosts": null,
        "GroupAdd": null,
        "IpcMode": "private",
        "Cgroup": "",
        "Links": null,
        "OomScoreAdj": 0,
        "PidMode": "",
        "Privileged": false,
        "PublishAllPorts": false,
        "ReadonlyRootfs": false,
        "SecurityOpt": null,
        "UTSMode": "",
        "UsernsMode": "",
        "ShmSize": 67108864,
        "Runtime": "runc",
        "Isolation": "",
        "CpuShares": 0,
        "Memory": 0,
        "NanoCpus": 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/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/overlay2/05b5d27e7b485efa8
1832c7dfbe02b1b4dd6e629cb2e0e1618d4926edac8e073/diff:/var/lib/docker/overlay2/ca28a1f5e1fa0628ee7b2499627f20a7ef0bf407f5d2b3bb84d94de
f6aedc13b/diff:/var/lib/docker/overlay2/40a1831b8c911e6ffc6462d5c5576d793d53cd4aecc3f681f3435d0db7567e09/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": {},
```

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)



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)



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

```
windarlobogarces909@AdrianPC:~/examen$ docker run -d -p 8080:80 --name daw-web daw-web:v1.0.0
607e5abc486731500e2fb769b4c6bf50d20c9ecf3fc9e2a4bcff311e1327e99b
windarlobogarces909@AdrianPC:~/examen$ docker network create dawnet
41f2eaf53566d4985cb2fea226c3e515f113254047256e286167e2e731d6ab1f
windarlobogarces909@AdrianPC:~/examen$ docker network connect dawnet daw-web
windarlobogarces909@AdrianPC:~/examen$
```

```
^Z^Cwindarlobogarces909@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)

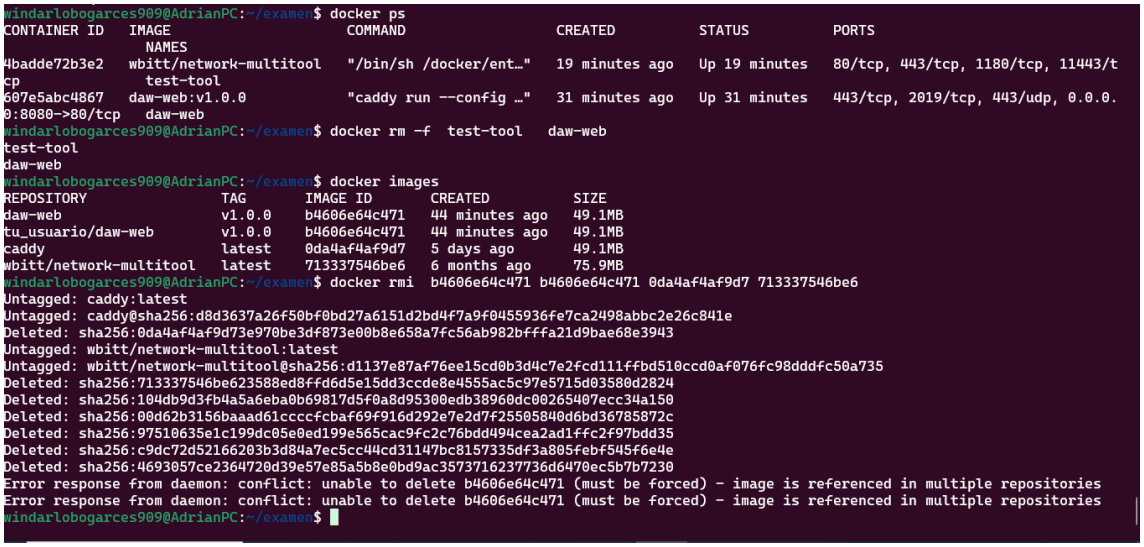
```
windarlobogarces909@AdrianPC:~/examen$ docker exec -it test-tool bash
4badde72b3e2:/# ping 192.168.128.2
PING 192.168.128.2 (192.168.128.2) 56(84) bytes of data.
```

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)

```
windarlobogarces909@AdrianPC:~/examen$ docker tag daw-web:v1.0.0 tu_usuario/daw-web:v1.0.0
windarlobogarces909@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
windarlobogarces909@AdrianPC:~/examen$
```

1.8. Elimina todos los contenedores, redes, volúmenes



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

- Accesible desde el puerto 8080 del anfitrión

Todos lo

s servicios deben conectarse a una red llamada “composenet”

docker-compose.yml

```yaml
version: "3.8"
services:
  mysql:
    image: mysql
    volumes:
      - dbdata:/var/lib/mysql
      - "./world-db:/docker-entrypoint-initdb.d"
    environment:
      MYSQL_DATABASE: world
      MYSQL_USER: root
      MYSQL_PASSWORD: example
      MYSQL_ROOT_PASSWORD: example
    ports:
      - "3306:3306"
    networks:
      - composenet

  phpmyadmin:
    image: phpmyadmin/phpmyadmin
    environment:
      PMA_HOST: mysql
      PMA_USER: root
      PMA_PASSWORD: example
    ports:
      - "8080:80"
    depends_on:
```

docker-compose.yml

```yaml
services:
  mysql:
    networks:

  phpmyadmin:
    image: phpmyadmin/phpmyadmin
    environment:
      PMA_HOST: mysql
      PMA_USER: root
      PMA_PASSWORD: example
    ports:
      - "8080:80"
    depends_on:
      - mysql
    networks:
      - composenet

networks:
  composenet:
    driver: bridge

volumes:
  dbdata:
    driver: local
```