

Levantamos un contenedor con la imagen de ubuntu y de forma interactiva con el comando “**docker run -it ubuntu /bin/bash**”.

```
root@mj_ubuntu_server:~# docker run -it ubuntu /bin/bash
root@6b0c336a8aed:/#
```

Dentro del contenedor, actualizamos los repositorios con “**apt-get update**”.

```
root@6b0c336a8aed:/# apt-get update
```

Luego, instalamos proftpd con “**apt-get install proftpd**”.

```
root@6b0c336a8aed:/# apt-get install proftpd
```

Instalamos además systemctl con “**apt install systemctl**”.

```
root@6b0c336a8aed:/# apt install systemctl
```

Instalamos un editor, que yo he elegido nano, con “**apt-get install nano**”.

```
root@6b0c336a8aed:/# apt-get install nano
```

Y procedemos a configurar el archivo proftpd.conf del contenedor de la siguiente manera:

```
root@6b0c336a8aed:/# nano /etc/proftpd/proftpd.conf
```

Apartado A.

Hacemos que los usuarios estén enjaulados en su Home descomentando la línea del “**DefaultRoot ~**”.

```
# Use this to jail all users in their homes
DefaultRoot ~
```

Apartado B.

Para que se permita la conexión con usuarios que no tengan una shell válida, descomentamos la línea “**RequireValidShell off**” y quedará así:

```
# Users require a valid shell listed in /etc/shells to login.
# Use this directive to release that constrain.
RequireValidShell      off
```

Apartado C.

Primero activando el anónimo desbloqueando todo su bloque, de forma que deberá quedar así:

```
GNU nano 2.9.3 /etc/proftpd/proftpd.conf

# A basic anonymous configuration, no upload directories.

<Anonymous ~ftp>
  User ftp
  Group nogroup
  # We want clients to be able to login with "anonymous" as well as "ftp"
  UserAlias anonymous ftp
  # Cosmetic changes, all files belongs to ftp user
  DirFakeUser on ftp
  DirFakeGroup on ftp

  RequireValidShell off

  # Limit the maximum number of anonymous logins
  MaxClients 10

  # We want 'welcome.msg' displayed at login, and '.message' displayed
  # in each newly chdired directory.
  DisplayLogin welcome.msg
  DisplayChdir .message

  # Limit WRITE everywhere in the anonymous chroot
  <Directory *>
    <Limit WRITE>
      DenyAll
    </Limit>
  </Directory>

  </Directory>

  # Uncomment this if you're brave.
  # <Directory incoming>
  #   # Umask 022 is a good standard umask to prevent new files and dirs
  #   # (second parm) from being group and world writable.
  #   Umask 022 022
  #   <Limit READ WRITE>
  #     DenyAll
  #   </Limit>
  #   <Limit STOR>
  #     AllowAll
  #   </Limit>
  # </Directory>

</Anonymous>

# Include other custom configuration files
Include /etc/proftpd/conf.d/
```

Una vez descomentado, para que solo pueda leer, modificamos el bloque de “<Directory *>” donde pondremos que se le denieguen todos los permisos y que el permiso de leer se le permita de la siguiente forma:

```
<Directory * >
  <Limit * >
    DenyAll
  </Limit>

  <Limit READ >
    AllowAll
  </Limit>
</Directory>
```

```
# Limit WRITE everywh
<Directory *>
  <Limit *>
    DenyAll
  </Limit>

  <Limit READ>
    AllowAll
  </Limit>
</Directory>
```

Por último, especificamos su DefaultRoot poniéndolo al lado de la etiqueta de Anonymous.

```
<Anonymous /srv/ftp>
  User
  Group
  # --
```

Apartado D.

Nos aseguramos que las siguientes directivas estén el archivo “**proftpd.conf**”:

- **Include /etc/proftpd/modules.conf**

```
# Includes DSO modules
Include /etc/proftpd/modules.conf
```

- **DefaultRoot ~**

```
# Use this to jail all users in their homes
DefaultRoot ~
```

- **RequireValidShell off**

```
# Users require a valid shell listed in /etc/shells to login.
# Use this directive to release that constrain.
RequireValidShell off
```

- **AuthUserFile /etc/proftpd/ftpd.passwd**

```
#Incluye el archivo de usuarios virtuales
AuthUserFile /etc/proftpd/ftpd.passwd
```

Una vez guardado el archivo proftpd.conf, salimos de él y creamos la carpeta **datos** y dentro de esta una carpeta por cada usuario virtual que vayamos a crear:

```
root@6b0c336a8aed:/# mkdir /datos
root@6b0c336a8aed:/# mkdir /datos/ftp_mara
root@6b0c336a8aed:/# mkdir /datos/ftp_eriku
root@6b0c336a8aed:/# mkdir /datos/ftp_oscaru
root@6b0c336a8aed:/#
```

Luego, le daremos permisos a las carpetas con los siguientes comandos:

- **chown ftp.nogroup /datos/ftp_mara /datos/ftp_eriku /datos/ftp_oscaru**
- **chmod 777 /datos/ftp_mara /datos/ftp_eriku /datos/ftp_oscaru**

```
root@6b0c336a8aed:/# chown ftp.nogroup /datos/ftp_mara /datos/ftp_eriku /datos/ftp_oscaru
root@6b0c336a8aed:/# chmod 777 /datos/ftp_mara /datos/ftp_eriku /datos/ftp_oscaru
root@6b0c336a8aed:/#
```

A continuación creamos el archivo donde guardaremos los usuarios virtuales y sus contraseñas con “**touch /etc/proftpd/ftpd.passwd**”.

```
root@6b0c336a8aed:/# touch /etc/proftpd/ftpd.passwd
root@6b0c336a8aed:/#
```

Nos posicionamos en la carpeta /etc/proftpd y creamos a los usuarios con el comando “**ftpasswd --passwd --name=nombre --uid=2000 --gid=2000 --home=/datos/ftp_nombre --shell=/bin/false**”.

Cuando los creamos nos pedirá su contraseña. Yo les he puesto a todos la contraseña de “**MJ**”. Para ello entonces hacemos:

- **ftpasswd --passwd --name=mara --uid=2000 --gid=2000 --home=/datos/ftp_mara --shell=/bin/false**

```
root@6b0c336a8aed:/etc/proftpd# ftpasswd --passwd --name=mara --uid=2000 --gid=2000 --home=/datos/ftp_mara --shell=/bin/false
ftpasswd: creating passwd entry for user mara

ftpasswd: /bin/false is not among the valid system shells. Use of
ftpasswd: "RequireValidShell off" may be required, and the PAM
ftpasswd: module configuration may need to be adjusted.

Password:
Re-type password:

ftpasswd: entry created
root@6b0c336a8aed:/etc/proftpd#
```

- `ftpasswd --passwd --name=eriku --uid=2001 --gid=2001 --home=/datos/ftp_eriku --shell=/bin/false`

```
ftpasswd: entry created
root@6b0c336a8aed:/etc/proftpd# ftppasswd --passwd --name=eriku --uid=2001 --gid=2001 --home=/datos/ftp_eriku --shell=/bin/false
ftpasswd: creating passwd entry for user eriku
```

- `ftpasswd --passwd --name=oscaru --uid=2002 --gid=2002 --home=/datos/ftp_oscaru --shell=/bin/false`

```
root@6b0c336a8aed:/etc/proftpd# ftppasswd --passwd --name=oscaru --uid=2002 --gid=2002 --home=/datos/ftp_oscaru --shell=/bin/false
ftpasswd: creating passwd entry for user oscaru
```

Y para comprobar que tenemos los usuarios creados en el archivo hacemos un “`cat ftpd.passwd`”.

```
root@6b0c336a8aed:/etc/proftpd# cat ftpd.passwd
mara:$1$MTZZEnmB$crH4oPBk0s0CiOfRPwppR/:2000:2000::/datos/ftp_mara:/bin/false
eriku:$1$1kJNu58p$knc0Sf/tTn5Bkv/wn64DI/:2001:2001::/datos/ftp_eriku:/bin/false
oscaru:$1$wkyygZFn$nlKPp9anAnlvOh3x.mDFN1:2002:2002::/datos/ftp_oscaru:/bin/false
root@6b0c336a8aed:/etc/proftpd#
```

Apartado E.

Solo configuraremos los puertos pasivos, de forma que descomentamos la línea “**PassivePorts**” y ponemos el rango especificado quedando “**PassivePorts 49000 49099**”.

```
# In some cases you have to specify passive ports range to by-pass
# firewall limitations. Ephemeral ports can be used for that, but
# feel free to use a more narrow range.
PassivePorts 49000 49099
```

Apartado F.

Ya terminado, salimos del contenedor dejándolo activo para probar que las configuraciones hechas hasta ahora han servido. Para salir del contenedor sin pararlo utilizamos “**Ctrl + P**” seguido de “**Ctrl + Q**”.

```
root@6b0c336a8aed:/etc/proftpd# nano /etc/proftpd/proftpd.conf
root@6b0c336a8aed:/etc/proftpd# root@mj_ubuntu_server:~#
root@mj_ubuntu_server:~#
```

Luego, vamos a saber cuál es la IP del contenedor. Para ello hacemos primero un “**docker ps**” para asegurarnos de que la máquina está corriendo.

```
root@mj_ubuntu_server:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
6b0c336a8aed   ubuntu   "/bin/bash"             14 minutes ago Up 14 minutes          festive_rhodes
```


Luego, miramos las redes que hay con “**docker network ls**”.

```
root@mj_ubuntu_server:~# docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
ec0a49035cc6        bridge             bridge             local
8153c11b4139        host              host              local
c40e2d56c5db        node_default       bridge             local
0142125a213d        none              null              local
fd7b93b29bf3        opencart-network   bridge             local
bla6dc11d11e        opencartNetwork    bridge             local
```

Ahora, buscamos la IP en una de las redes que aparecen. Normalmente, los contenedores usan por defecto la red “**bridge**”. Si usamos otra que hemos especificado al crear el contenedor, deberemos buscarla en el nombre de dicha red especificada. Para buscar dentro utilizamos el comando “**docker network inspect idRed**”. En nuestro caso, la ID de bridge empieza por “ec0”, así que utilizamos eso.

```
bla6dc11d11e    opencartNetwork    bridge    local
root@mj_ubuntu_server:~# docker network inspect ec0
```

Y dentro, buscamos en la zona de “**containers**” uno que tenga el nombre/id del contenedor que necesitamos. Nosotros buscamos el contenedor de nombre “**festive_rhodes**” y vemos que su IP es **172.17.0.2**.

```
},
"ConfigOnly": false,
"Containers": {
  "6b0c336a8aedd39aafdcle7818cffc9075f6501dfd922db0556c0a1167bddb9e": {
    "Name": "festive_rhodes",
    "EndpointID": "a6b2db3e8a6a68794802aeb40a5c0b9a24c9951672b133a783ef288d0e748757",
    "MacAddress": "02:42:ac:11:00:02",
    "IPv4Address": "172.17.0.2/16",
    "IPv6Address": ""
  }
}
}.
```

Una vez sabida la IP, entramos en el contenedor de nuevo para activar el servicio proftpd a mano y así hacer las pruebas de funcionamiento. Lo pondremos en marcha con “**systemctl status proftpd**”.

```
root@mj_ubuntu_server:~# docker exec -it 6b0 /bin/bash
root@6b0c336a8aed:/# systemctl start proftpd
root@6b0c336a8aed:/# systemctl status proftpd
proftpd.service - This script runs the FTP service offered
   Loaded: loaded (/etc/init.d/proftpd, disabled)
   Active: active (running)
root@6b0c336a8aed:/#
```

Y salimos con “**Ctrl + P**” y “**Ctrl + Q**”. Luego, accedemos a ftp con la IP del contenedor y probamos a poner el usuario “**mara**” de contraseña “**MJ**” y vemos que accede.

```

root@mj_ubuntu_server:~# ftp
ftp> open 172.17.0.2
Connected to 172.17.0.2.
220 ProFTPD Server (Debian) [::ffff:172.17.0.2]
Name (172.17.0.2:mj_ubuntu_server): mara
331 Password required for mara
Password:
230 User mara logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>

```

Una vez comprobado que la configuración funciona, creamos una imagen a partir del contenedor utilizando el comando “**docker commit nombreContenedor/idContenedor nombreNuevaImagen**”. Yo utilizaré “**docker commit 6b0 silecsis/mjcg_proftpd**”.

```

root@mj_ubuntu_server:~# docker commit 6b0 silecsis/mjcg_proftpd
sha256:616b27091574c5bc7dccbdbf3394c3a825907e89b700c3a08bdc3298425fb6b2
root@mj_ubuntu_server:~#

```

Y luego lo subimos a nuestro docker hub. Primero nos logueamos:

```

root@mj_ubuntu_server:~# docker login -u silecsis
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
root@mj_ubuntu_server:~#

```

Y luego lo subimos al repositorio:

```

root@mj_ubuntu_server:~# docker push silecsis/mjcg_proftpd
Using default tag: latest
The push refers to repository [docker.io/silecsis/mjcg_proftpd]
4145584e282c: Pushing 2.173MB/72.24MB

```

Comprobamos que se haya subido en el docker hub:

silecsis / mjcg_proftpd Updated a few seconds ago	Not Scanned	☆ 0	↓ 18	Public
silecsis / mysql Updated 2 days ago	Not Scanned	☆ 0	↓ 1	Public
silecsis / php5 Updated 8 days ago	Not Scanned	☆ 0	↓ 4	Public
silecsis / apache2 Updated 8 days ago	Not Scanned	☆ 0	↓ 11	Public

NOTA: Si falla el loguearse.

Si al loguearnos nos da un error del “daemon”:

```
root@mj_ubuntu_server:~# docker login -u silecsis
Password:
Error response from daemon: Get https://registry-1.docker.io/v2/: unauthorized: incorrect username or password
root@mj_ubuntu_server:~#
```

Actualizamos los repositorios con “**apt-get update**”.

```
root@mj_ubuntu_server:~# apt update
```

Y luego introducimos el siguiente comando: “**apt -V install gnupg2 pass**”.

```
root@mj_ubuntu_server:~# apt -V install gnupg2 pass
Reading package lists... Done
Building dependency tree
Reading state information... Done
pass is already the newest version (1.7.1-3).
```

Luego podremos volver a loguearnos tranquilamente porque funcionará.

Apartado G.

Creamos dos volúmenes, uno para **/datos** y otro para **/srv/ftp**.

```
root@mj_ubuntu_server:~# docker volume create carpeta_datos
carpeta_datos
root@mj_ubuntu_server:~# docker volume create carpeta_srv_ftp
carpeta_srv_ftp
root@mj_ubuntu_server:~#
```

Apartado H.

Ahora deberemos levantar un contenedor de forma que utilice la imagen creada **silecsis/mjcg_proftpd**, los volúmenes creados **carpeta_datos** y **carpeta_srv_ftp**. Que además se quede **funcionando en segundo plazo** y que tenga el **servicio activado de proftpd**.

Para realizar todo esto insertamos el siguiente comando:

“**docker run -d -it --name prueba_proftpd -v carpeta_datos:/datos -v carpeta_srv_ftp:/srv/ftp silecsis/mjcg_proftpd systemctl start proftpd /bin/bash**”

```
carpeta_srv_ftp
root@mj_ubuntu_server:~# docker run -d -it --name prueba_proftpd -v carpeta_datos:/datos -v carpeta_srv_ftp:/srv/ftp silecsis/mjcg_proftpd systemctl start proftpd /bin/bash
c5d939cf4353890d1153b0c2d4dbc72b4b942484226ff05807bda6ed326640cf
root@mj_ubuntu_server:~#
```

Donde:

- ‘-d’: Inicia el contenedor en 2º plano.
- ‘-it’ y ‘/bin/bash’: Nos abre el shell del contenedor y lo inicia.
- ‘--name prueba_proftpd’: El comando ‘--name’ pone nombre al contenedor y ‘prueba_proftpd’ es el nombre otorgado.
- ‘-v carpeta_datos:/datos’: llamará al volumen y seguido pondremos la ruta que queremos que guarde.
- ‘silecsis/mjcg_proftpd’: Será la imagen de la cual queremos crear el contenedor.
- ‘systemctl start proftpd’: Será el comando necesario para inicializar el servicio proftpd cuando levantemos el contenedor.

Luego comprobamos que se haya levantado iniciado con el comando “**docker ps**”:

```
root@mj_ubuntu_server:~# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
c5d939cf4353	silecsis/mjcg_proftpd	"systemctl start pro..."	19 seconds ago	Up 18 seconds	

```
prueba_proftpd
```

Apartado I.

Ahora realizaremos los mismos pasos explicados anteriormente para saber la IP. Primero hacemos un ls de las redes para saber la id para inspeccionarla.

```
festive_rhodes
root@mj_ubuntu_server:~# docker network ls
```

NETWORK ID	NAME	DRIVER	SCOPE
ec0a49035cc6	bridge	bridge	local
8153c11b4139	host	host	local
c40e2d56c5db	node_default	bridge	local
0142125a213d	none	null	local
fd7b93b29bf3	opencart-network	bridge	local
bla6dc11d11e	opencartNetwork	bridge	local

```
root@mj_ubuntu_server:~# docker network inspect ec0
```

Luego la inspeccionamos y vemos la IP que es **172.17.0.3**.

```
IPV4ADDRESS : 172.17.0.2/16 ,
"IPv6Address": ""
},
"c5d939cf4353890d1153b0c2d4dbc72b4b942484226ff05807bda6ed326640cf": {
  "Name": "prueba_proftpd",
  "EndpointID": "8ed24239518b29c8eb2ef98e471ba2a6e3218fac32fa7ae741a7bfdda6a4e3b4",
  "MacAddress": "02:42:ac:11:00:03",
  "IPv4Address": "172.17.0.3/16",
  "IPv6Address": ""
}
```

Y probamos la configuración FTP. Primero probamos a que se pueda conectar con uno de los usuarios virtuales.

```

root@mj_ubuntu_server:~# ftp 172.17.0.3
Connected to 172.17.0.3.
220 ProFTPD Server (Debian) [::ffff:172.17.0.3]
Name (172.17.0.3:mj_ubuntu_server): mara
331 Password required for mara
Password:
230 User mara logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>

```

(**NOTA:** sale una IP diferente porque las pruebas las hice después y borré un contenedor que antes estaba en funcionamiento, por eso cambia a esa IP. Pero es el mismo contenedor.)

En esta prueba pruebo a conectarme al anónimo . Y dentro, hago un “ls” viendo los permisos que tiene, que son solo de lectura.

```

root@mj_ubuntu_server:~# ftp 172.17.0.2
Connected to 172.17.0.2.
220 ProFTPD Server (Debian) [::ffff:172.17.0.2]
Name (172.17.0.2:mj_ubuntu_server): anonymous
331 Anonymous login ok, send your complete email address as your password
Password:
230-Welcome, archive user anonymous@172.17.0.1 !
230-
230-The local time is: Fri Feb 12 08:56:39 2021
230-
230-This is an experimental FTP server. If you have any unusual problems,
230-please report them via e-mail to <root@c5d939cf4353>.
230-
230 Anonymous access granted, restrictions apply
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful
150 Opening ASCII mode data connection for file list
-rw-r--r-- 1 ftp ftp 170 Feb 27 2020 welcome.msg
226 Transfer complete
ftp>

```

Comando MasqueradeAddress.

Si necesitáramos realizar una comprobación desde el Filezilla del anfitrión, deberemos habilitar dicha directiva (MasqueradeAddress) con la IP de la máquina virtual.

Para no tocar el archivo de proftpd.conf, lo haremos al levantar el contenedor con un parámetro: “MASQ_ADDR”. Éste tendrá que ir seguido de la IP de la máquina, es decir, “-e MASQ_ADDR:192.168.150.8”.

Además, para que funcione correctamente, deberemos ponerle una conexión de red mediante la red “host”. Es decir, el comando al levantar la máquina también tendrá que tener “--net=host”.

Por lo tanto, yo he levantado otro contenedor con la misma imagen y los mismos parámetros de antes pero con esta directiva con el comando **“docker run -d -it --net=host --name prueba_dos -v carpeta_datos:/datos -v carpeta_srv_ftp:/srv/ftp -e MASQ_ADDR:192.168.150.8 silecsis/mjcg_proftpd systemctl start proftpd /bin/bash”**.

```
root@mj_ubuntu_server:~# docker run -dit --net=host --name prueba_dos -v carpeta_datos:/datos -v carpeta_srv_ftp:/srv/ftp -e MASQ_ADDR:192.168.150.8 silecsis/mjcg_proftpd systemctl start proftpd /bin/bash
3a4d8dc77158eee6dbcd7197e4dcabd21fdf8e551f72f3a39e50651f0f0c2a81
root@mj_ubuntu_server:~#
```

Luego, entramos en el Filezilla y metemos la IP de la máquina virtual entrando con el usuario y contraseña del contenedor. Podemos ver que conecta perfectamente.

