

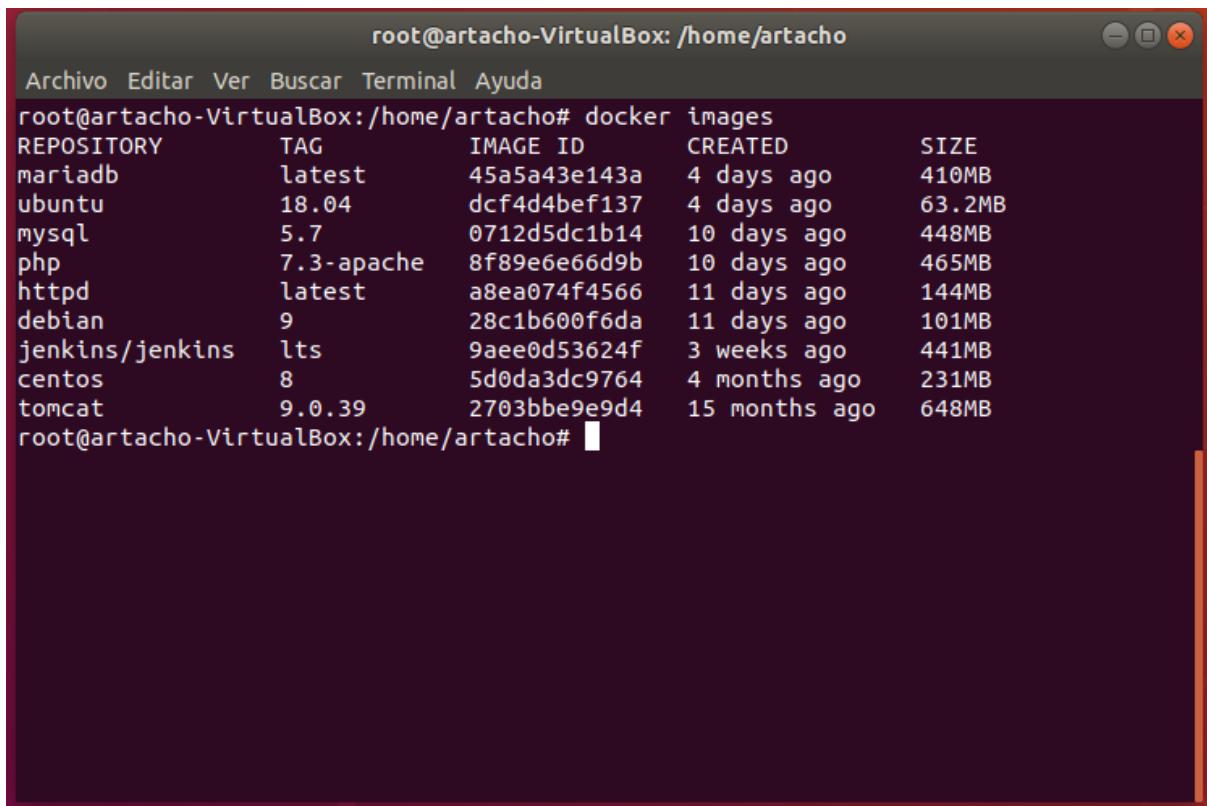
Tarea. Descarga de imágenes (OBLIGATORIA)

Debes descargar las siguientes imágenes:

- ubuntu:18.04
- centos:8
- debian:9
- mariadb:latest
- mysql:5.7
- httpd
- tomcat:9.0.39
- jenkins/jenkins:lts
- php:7.3-apache

Una vez hayas descargado todas las imágenes, ejecuta la orden de Docker cli correspondiente que compruebe que se han descargado y realiza un pantallazo del resultado.

Para descargar cualquier imagen usamos el comando “**docker pull nombre_imagen:versión**”



```
root@artacho-VirtualBox: /home/artacho
root@artacho-VirtualBox: /home/artacho# docker images
REPOSITORY      TAG          IMAGE ID   CREATED     SIZE
mariadb        latest       45a5a43e143a  4 days ago  410MB
ubuntu         18.04       dcf4d4bef137  4 days ago  63.2MB
mysql          5.7         0712d5dc1b14  10 days ago 448MB
php             7.3-apache  8f89e6e66d9b  10 days ago 465MB
httpd           latest       a8ea074f4566  11 days ago 144MB
debian          9            28c1b600f6da  11 days ago 101MB
jenkins/jenkins lts          9aee0d53624f  3 weeks ago 441MB
centos          8            5d0da3dc9764  4 months ago 231MB
tomcat          9.0.39      2703bbe9e9d4  15 months ago 648MB
root@artacho-VirtualBox: /home/artacho#
```

Tarea. Arranca un contenedor (OBLIGATORIA)

En esta tarea vas a arrancar un contenedor al que llamaremos ubuntu. Arrancar un contenedor de la imagen ubuntu:18.04 al que llamaremos ubuntu. Una vez arrancado realizar las siguiente operaciones:

1. Salir del contenedor y capturar un pantallazo donde se pueda comprobar que ese contenedor se ha parado.
2. Rearrancar el contenedor desde tu equipo y capturar un pantallazo donde se pueda comprobar que está funcionando.
3. Sin entrar en el contenedor, mostrar por pantalla el fichero /etc/os-release y capturar un pantallazo del resultado.

Para crear un contenedor usamos el comando “**docker run –name nombre_contenedor -it nombre_imagen**” (la variable de entorno -it se usa para acceder al shell)

Para iniciar el contenedor usamos el comando “**docker start nombre_contenedor**” (podemos añadir la variable -i si queremos acceder al shell)

Para detener un contenedor usamos el comando “**docker stop nombre_contenedor**”

```
root@artacho-VirtualBox:/home/artacho# docker run --name ubuntu -it ubuntu:18.04
root@7e4b56a23efa:# exit
exit
root@artacho-VirtualBox:/home/artacho# docker ps -a
CONTAINER ID   IMAGE      COMMAND   CREATED      STATUS          PORTS     NAMES
7e4b56a23efa   ubuntu:18.04 "bash"    2 minutes ago   Exited (0) About a minute ago
root@artacho-VirtualBox:/home/artacho# docker stop ubuntu
ubuntu
root@artacho-VirtualBox:/home/artacho# docker ps
CONTAINER ID   IMAGE      COMMAND   CREATED      STATUS          PORTS     NAMES
root@artacho-VirtualBox:/home/artacho# docker start ubuntu
ubuntu
root@artacho-VirtualBox:/home/artacho# docker ps
CONTAINER ID   IMAGE      COMMAND   CREATED      STATUS          PORTS     NAMES
7e4b56a23efa   ubuntu:18.04 "bash"    2 minutes ago   Up 7 seconds
root@artacho-VirtualBox:/home/artacho# cat /etc/os-release
NAME="Ubuntu"
VERSION="18.04.6 LTS (Bionic Beaver)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 18.04.6 LTS"
VERSION_ID="18.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=bionic
UBUNTU_CODENAME=bionic
root@artacho-VirtualBox:/home/artacho# █
```

En esta tareas vas a ejecutar varios servicios web y de bases de datos sobre contenedores.

Debes realizar las siguientes acciones:

1. Arranca un contenedor que ejecute una instancia de la imagen php:7.3-apache, que se llame web y que sea accesible desde tu equipo en el puerto 8181.
2. Colocar en el directorio raíz del servicio web de dicho contenedor un fichero llamado index.html con el siguiente contenido:

```
<h1>HOLA SOY XXXXXXXXXXXXXXXX</h1>
```

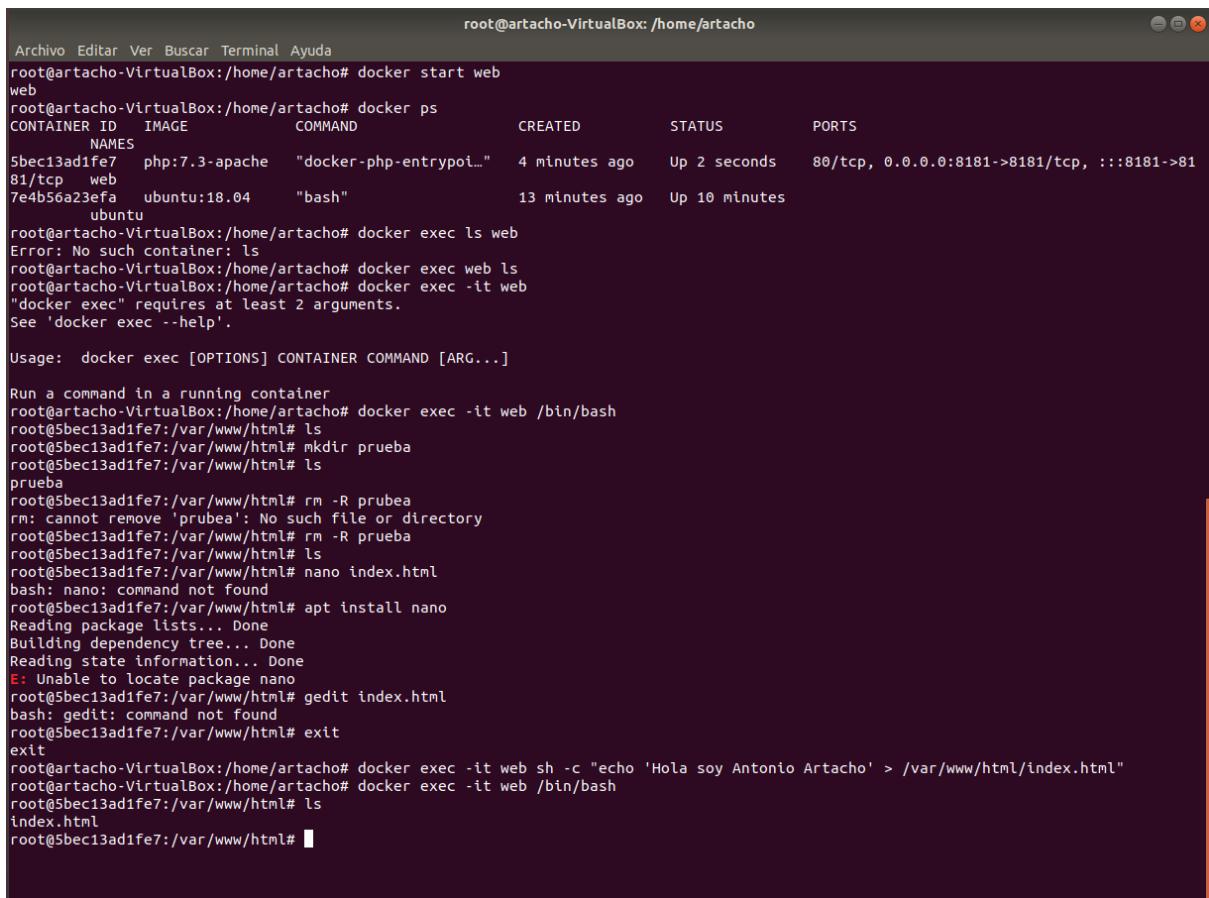
Deberás sustituir XXXXXXXXXXXXXXX por tu nombre y tus apellidos.

Para arrancar el contenedor en un puerto usamos el comando “**docker run –name nombre_contenedor -d -p puerto_host:puerto_contenedor versión_imagen**”

Para crear un archivo en el contenedor usamos el comando “**docker exec -it nombre_contenedor sh -c “echo ‘Contenido del archivo’ > ruta/_nombre_archivo”**”

Para acceder al shell usamos el comando “**docker exec nombre_contenedor -it /bin/bash**”

```
root@artacho-VirtualBox:/home/artacho# docker run --name web -d -p 8181:8181 php:7.3-apache
5bec13ad1fe76a1c1f115281eaa8af696e4ac087459826114e555f62bf453a81
root@artacho-VirtualBox:/home/artacho# docker start web
```



The screenshot shows a terminal window with the following session:

```
root@artacho-VirtualBox:/home/artacho#
Archivo Editar Ver Buscar Terminal Ayuda
root@artacho-VirtualBox:/home/artacho# docker start web
web
root@artacho-VirtualBox:/home/artacho# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
 NAMES
5bec13ad1fe7 php:7.3-apache "docker-php-entrypoi..." 4 minutes ago Up 2 seconds 80/tcp, 0.0.0.0:8181->8181/tcp, :::8181->8181/tcp web
7e4b56a23ef9 ubuntu:18.04 "bash"
13 minutes ago Up 10 minutes
ubuntu
root@artacho-VirtualBox:/home/artacho# docker exec ls web
Error: No such container: ls
root@artacho-VirtualBox:/home/artacho# docker exec web ls
root@artacho-VirtualBox:/home/artacho# docker exec -it web
"docker exec" requires at least 2 arguments.
See 'docker exec --help'.
Usage: docker exec [OPTIONS] CONTAINER COMMAND [ARG...]
Run a command in a running container
root@artacho-VirtualBox:/home/artacho# docker exec -it web /bin/bash
root@5bec13ad1fe7:/var/www/html# ls
root@5bec13ad1fe7:/var/www/html# mkdir prueba
root@5bec13ad1fe7:/var/www/html# ls
prueba
root@5bec13ad1fe7:/var/www/html# rm -R prueba
rm: cannot remove 'prueba': No such file or directory
root@5bec13ad1fe7:/var/www/html# rm -R prueba
root@5bec13ad1fe7:/var/www/html# ls
root@5bec13ad1fe7:/var/www/html# nano index.html
bash: nano: command not found
root@5bec13ad1fe7:/var/www/html# apt install nano
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package nano
root@5bec13ad1fe7:/var/www/html# gedit index.html
bash: gedit: command not found
root@5bec13ad1fe7:/var/www/html# exit
exit
root@artacho-VirtualBox:/home/artacho# docker exec -it web sh -c "echo 'Hola soy Antonio Artacho' > /var/www/html/index.html"
root@artacho-VirtualBox:/home/artacho# docker exec -it web /bin/bash
root@5bec13ad1fe7:/var/www/html# ls
index.html
root@5bec13ad1fe7:/var/www/html#
```

3. Colocar en ese mismo directorio raíz un archivo llamado index.php con el siguiente contenido:

```
root@artacho-VirtualBox:/home/artacho# docker exec -it web sh -c "echo '<?php phpinfo(); ?>' > /var/www/html/index.php"
root@artacho-VirtualBox:/home/artacho# docker exec -it web /bin/bash
root@5bec13ad1fe7:/var/www/html#
index.html index.php
root@5bec13ad1fe7:/var/www/html# exit
exit
root@artacho-VirtualBox:/home/artacho#
```

PHP Version 7.3.33

System	Linux 8f4aba51e656 5.4.0-97-generic #110~18.04.1-Ubuntu SMP Mon Jan 17 20:50:31 UTC 2022 x86_64
Build Date	Jan 26 2022 18:41:10
Configure Command	'./configure' '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--with-pic' '--enable-ftp' '--enable-mbstring' '--enable-mysqli' '--with-password-argon2' '--with-sodium=shared' '--with-pdo-sqlite=/usr' '--with-sqlite3=/usr' '--with-curl' '--with-openssl' '--with-readline' '--with-zlib' '--with-libdir=/lib/x86_64-linux-gnu' '--disable-cgi' '--with-apxs2' 'build_alias=x86_64-linux-gnu'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	(none)
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-sodium.ini
PHP API	20180731
PHP Extension	20180731
Zend Extension	320180731
Zend Extension Build	API320180731,NTS
PHP Extension Build	API20180731,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

This program makes use of the Zend Scripting Language Engine:
Zend Engine v3.3.33, Copyright (c) 1998-2018 Zend Technologies

zend engine



4. Arrancar un contenedor que se llame bbdd y que ejecute una instancia de la imagen mariadb para que sea accesible desde el puerto 3336. Antes de arrancarlo visitar la página del contenedor en Docker Hub (https://hub.docker.com/_/mariadb) y establecer las variables de entorno necesarias para que:

La contraseña de root sea root.

Crear una base de datos automáticamente al arrancar que se llame prueba.

Crear el usuario invitado con la contraseña invitado.

Deberás entregar los siguientes pantallazos comprimidos en un único fichero: o
Pantallazo que desde el navegador muestre el fichero index.html

```
root@artacho-VirtualBox: /home/artacho
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 5.4                                         index.html
<h1>Hola, soy Antonio Artacho</h1>
```

Pantallazo que desde el navegador muestre el fichero index.php

```
root@artacho-VirtualBox: /home/artacho
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 5.4                                         index.php
<?php phpinfo(); ?>
```

Pantallazo donde desde un cliente de base de datos se pueda observar que hemos podido conectarnos al servidor de base de datos con el usuario creado y que se ha creado la base de datos prueba (show databases).

```
root@artacho-VirtualBox: /home/artacho
Archivo Editar Ver Buscar Terminal Ayuda
root@artacho-VirtualBox:/home/artacho# docker run --detach --publish 3336:80 --name bddd --env MARIADB_DATABASE=prueba --env MARIADB_USER=Invitado --env MARIADB_PASSWORD=invitado --env MARIADB_ROOT_PASSWORD=root mariadb:latest
7190644c32d817fe0ded2ad632b2b013435677cc76ff2f210556ffce5db9e90
root@artacho-VirtualBox:/home/artacho# docker ps
CONTAINER ID   IMAGE          COMMAND       CREATED      STATUS      PORTS
 NAMES
7190644c32d8   mariadb:latest "docker-entrypoint.s..."  8 seconds ago  Up 7 seconds  3306/tcp, 0.0.0.0:3336->80/tcp, :::3336->80
/tcp
bbdd
8f4aba51e656   php:7.3-apache "docker-php-entrypoi..."  10 minutes ago Up 10 minutes  0.0.0.0:8181->80/tcp, :::8181->80/tcp
web
7e4b56a23efa   ubuntu:18.04   "bash"        54 minutes ago Up 51 minutes
ubuntu
root@artacho-VirtualBox:/home/artacho#
```

```
root@7190644c32d8:/# mysql -u invitado -pinvitado
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 6
Server version: 10.6.5-MariaDB-1:10.6.5+maria~focal mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| prueba |
+-----+
2 rows in set (0.000 sec)

MariaDB [(none)]>
```

Tarea. Obteniendo información de los contenedores

En esta tarea vamos a obtener información de los contenedores creados previamente en la Tarea anterior. Ejecutando servicios desde contenedores. Servidores web y de bases de datos.

En el siguiente apartado se detalla la hoja de ruta para resolver la actividad.

Al final de esta tarea se encuentran los recursos necesarios para su realización.

Partiendo de los contenedores en ejecución de la tarea anterior y ejecutando la orden de docker cli adecuada obtener la siguiente información realizando un pantallazo cada vez:

Dirección IP del contenedor web

```
root@artacho-VirtualBox:/home/artacho# docker inspect --format 'La ip es > {{.NetworkSettings.Networks.bridge.IPAddress}}' web
La ip es > 172.17.0.2
```

Redirección de puertos del contenedor web

```
root@artacho-VirtualBox:/home/artacho# docker inspect --format 'Las direcciones de puertos son > {{.NetworkSettings.Ports}}' web
Las direcciones de puertos son > map[80/tcp:[{0.0.0.0 8181} :: 8181]]
root@artacho-VirtualBox:/home/artacho#
```

Dirección IP del contenedor bbdd

```
root@artacho-VirtualBox:/home/artacho# docker inspect --format 'La ip es > {{.NetworkSettings.Networks.bridge.IPAddress}}' bbdd
La ip es > 172.17.0.4
root@artacho-VirtualBox:/home/artacho#
```

Redirección de puertos del contenedor bbdd

```
root@artacho-VirtualBox:/home/artacho# docker inspect --format 'Las direcciones de puertos son > {{.NetworkSettings.Ports}}' bbdd
Las direcciones de puertos son > map[3306/tcp:[] 80/tcp:[{0.0.0.0 3336} :: 3336]]
root@artacho-VirtualBox:/home/artacho#
```

Tarea. Descarga, inspecciona y borra imágenes.

La gestión de las imágenes en Docker es un proceso sencillo con dos operaciones básicas, la descarga de las imágenes y su borrado.

En el siguiente apartado se detalla la hoja de ruta para resolver la actividad.

Al final de esta tarea se encuentran los recursos necesarios para su realización.

Para poder practicar la gestión de imágenes realizaremos las siguiente operaciones:

1. Descargar una nueva imagen desde Docker Hub, la imagen Ubuntu:20.04. Capturaremos un pantallazo donde usando el comando docker adecuado podamos comprobar que efectivamente esa imagen se ha descargado.

```
root@artacho-VirtualBox:/home/artacho# docker inspect --format 'Las direcciones de puertos son > {{.NetworkSettings.Ports}}' bbdd
Las direcciones de puertos son > map[3306/tcp:[] 80/tcp:[{0.0.0.0 3336} {:: 3336}]]
root@artacho-VirtualBox:/home/artacho# docker pull ubuntu:20.04
20.04: Pulling from library/ubuntu
08c01a0ec47e: Already exists
Digest: sha256:669e010b58baf5beb2836b253c1fd5768333f0d1dbc834f7c07a4dc93f474be
Status: Downloaded newer image for ubuntu:20.04
docker.io/library/ubuntu:20.04
root@artacho-VirtualBox:/home/artacho# docker images
REPOSITORY      TAG          IMAGE ID      CREATED       SIZE
mariadb         latest        45a5a43e143a   7 days ago   410MB
ubuntu          20.04        54c9d81cbb44   7 days ago   72.8MB
```

2. Obtener toda la información de esa imagen y volcarla el fichero info.txt.

```
root@artacho-VirtualBox:/home/artacho# docker image inspect ubuntu:20.04 > info.txt
root@artacho-VirtualBox:/home/artacho# ls
Descargas  Escritorio  Imágenes  Música  Públco
Documentos examples.desktop  info.txt  Plantillas  Vídeos
root@artacho-VirtualBox:/home/artacho# cat info.txt
[{"Id": "sha256:54c9d81cbb440897908abdcaa98674db83444636c300170cf211e40a66f704f",
 "RepoTags": [
     "ubuntu:20.04"
 ],
 "RepoDigests": [
     "ubuntu@sha256:669e010b58baf5beb2836b253c1fd5768333f0d1dbc834f7c07a4dc93f474be"
 ],
 "Parent": "",
 "Comment": "",
 "Created": "2022-02-02T02:14:46.177066251Z",
 "Container": "3d4cc5cf7dc1af55a2be4440b5be4f96ea35516b98407a9b9446c218bb43818a",
 "ContainerConfig": {
     "Hostname": "3d4cc5cf7dc1",
     "Domainname": "",
     "User": "",
     "AttachStdin": false,
     "AttachStdout": false,
     "AttachStderr": false,
     "Tty": false,
     "OpenStdin": false,
     "StdinOnce": false,
     "Env": [
         "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
     ]
 }
```

3. Instanciar esa imagen creando un contenedor llamado modulo3. Captura un pantallazo donde usando el comando docker adecuado podemos comprobar que efectivamente ese contenedor se ha creado (en ejecución o no).

```
root@artacho-VirtualBox:/home/artacho# docker run --name modulo3 ubuntu:20.04
root@artacho-VirtualBox:/home/artacho# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
2b6c6dad4a72      ubuntu:20.04       "bash"              20 seconds ago   Exited (0) 19 seconds ago
                                         modulo3
7190644c32d8      mariadb:latest    "docker-entrypoint.s..."   3 days ago       Exited (255) 28 minutes ago   3306/tcp, 0.0.0
.0:3336->80/tcp, :::3336->80/tcp   bddd
8f4aba51e656      php:7.3-apache   "docker-php-entrypoi..."   3 days ago       Up 16 minutes          0.0.0.0:8181->8
0/tcp, :::8181->80/tcp                web
7e4b56a23efa      ubuntu:18.04     "bash"              3 days ago       Exited (255) 28 minutes ago
                                         ubuntu
root@artacho-VirtualBox:/home/artacho#
```

4. Intentar borrar la imagen Ubuntu:20.04. ¿Has podido borrar la imagen? Responde razonadamente.

```
root@artacho-VirtualBox:/home/artacho# docker rmi ubuntu:20.04
Error response from daemon: conflict: unable to remove repository reference "ubuntu:20.04" (must force) - container 2b6
c6dad4a72 is using its referenced image 54c9d81cbb44
root@artacho-VirtualBox:/home/artacho#
```

No he podido borrar la imagen porque hay un contenedor (modulo3) que la está usando.

5. Realizar las operaciones necesarias para poder borrar la imagen. Captura un pantallazo de cada una de ellas.

```
root@artacho-VirtualBox:/home/artacho# docker rm modulo3
modulo3
root@artacho-VirtualBox:/home/artacho# docker rmi ubuntu:20.04
Untagged: ubuntu:20.04
Untagged: ubuntu@sha256:669e010b58baf5beb2836b253c1fd5768333f0d1dbc834f7c07a4dc93f474be
Deleted: sha256:54c9d81ccb440897908abdcaa98674db83444636c300170cf211e40a66f704f
root@artacho-VirtualBox:/home/artacho#
```

6. Insertar todos los pantallazos, junto con la respuesta a la pregunta formulada en un documento de texto llamado modulo3_docker (formatos permitidos odt/doc).

7. Comprimir ese documento junto con el info.txt y entregarlo.

Tarea. Creación y uso de volúmenes.

- 1.- Crear los siguientes volúmenes con la orden docker volume: volumen_datos
volumen_web

```
Archivo Editar Ver Buscar Terminal Ayuda
root@artacho-VirtualBox:/home/artacho# docker volume create volumen_datos
volumen_datos
root@artacho-VirtualBox:/home/artacho# docker volume create volumen_web
volumen_web
root@artacho-VirtualBox:/home/artacho#
```

```
root@artacho-VirtualBox:/home/artacho# docker volume ls
DRIVER      VOLUME NAME
local      b4dd14889e6a976e7e9df62bb12598a043077779936848dcb49904967ffa4b1a
local      volumen_datos
local      volumen_web
root@artacho-VirtualBox:/home/artacho#
```

2.- Una vez creados estos volumenes: Arrancar un contenedor llamado c1 sobre la imagen php:7.4- apache que monte el volumen_web en la ruta /var/www/html Arrancar un contenedor llamado c2 sobre la imagen mariadb que monte el volumen_datos en la ruta /var/lib/mysql y cuya contraseña de root sea admin.

```
root@artacho-VirtualBox:/home/artacho# docker run --name c1 --mount type=volume,src=volumen_web,dst=/var/www/html php:7.4-apache
Unable to find image 'php:7.4-apache' locally
7.4-apache: Pulling from library/php
5eb5b503b376: Already exists
8b1ad84cf101: Already exists
38c937dadeb7: Already exists
6a2f1dc96e59: Already exists
f8c3f82c39d4: Already exists
90fc6462bd8e: Already exists
c670d99116c9: Already exists
12c48eb41a93: Pull complete
0c537b2632b2: Pull complete
21a949b7ce7d: Pull complete
6e7c48790426: Pull complete
e1b625ec6dbf: Pull complete
345e72046e34: Pull complete
Digest: sha256:23d69eeef4f12618dea1202e38b345591ff04dc7faedd5b2c48b1d973eb04269
Status: Downloaded newer image for php:7.4-apache
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.4. Set the 'ServerName' directive globally to suppress this message
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.4. Set the 'ServerName' directive globally to suppress this message
[Wed Feb 09 17:52:57.229650 2022] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.52 (Debian) PHP/7.4.27 configured -- resuming normal operations
[Wed Feb 09 17:52:57.229824 2022] [core:notice] [pid 1] AH00094: Command line: 'apache2 -D FOREGROUND'
```

```
root@artacho-VirtualBox:/home/artacho# docker run --name c2 --mount type=volume,src=volumen_datos,dst=/var/lib/mysql -e MARIADB_ROOT_PASSWORD=admin mariadb:latest
2022-02-09 17:57:51+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:10.6.5+maria~focal started.
2022-02-09 17:57:52+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2022-02-09 17:57:52+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:10.6.5+maria~focal started.
2022-02-09 17:57:52+00:00 [Note] [Entrypoint]: Initializing database files
2022-02-09 17:57:52 0 [Warning] You need to use --log-bin to make --expire-logs-days or --binlog-expire-logs-seconds work.

PLEASE REMEMBER TO SET A PASSWORD FOR THE MariaDB root USER !
To do so, start the server, then issue the following command:

'/usr/bin/mysql_secure_installation'

which will also give you the option of removing the test
databases and anonymous user created by default. This is
strongly recommended for production servers.

See the MariaDB Knowledgebase at https://mariadb.com/kb or the
MySQL manual for more instructions.

Please report any problems at https://mariadb.org/jira

The latest information about MariaDB is available at https://mariadb.org/.
You can find additional information about the MySQL part at:
https://dev.mysql.com
Consider joining MariaDB's strong and vibrant community:
https://mariadb.org/get-involved/
```

3.- Parar y borrar el contenedor c2 y tras ello borrar el volumen volumen_datos.

```
root@artacho-VirtualBox:/home/artacho# docker stop c2
c2
root@artacho-VirtualBox:/home/artacho# docker rm c2
c2
root@artacho-VirtualBox:/home/artacho# docker volume rm volumen_datos
volumen_datos
root@artacho-VirtualBox:/home/artacho#
```

```
root@artacho-VirtualBox:/home/artacho# docker inspect c1
[{"Id": "2ba11810c01d8ef3cdfb0d8427883a9136d813d70a1233d4e93c5ceee0d90e43", "Created": "2022-02-09T17:52:55.349028159Z", "Path": "docker-php-entrypoint", "Args": ["apache2-foreground"], "State": {"Status": "exited", "Running": false, "Paused": false, "Restarting": false, "OOMKilled": false, "Dead": false, "Pid": 0, "ExitCode": 0, "Error": "", "StartedAt": "2022-02-09T17:52:56.81961616Z", "FinishedAt": "2022-02-09T17:56:24.676245041Z"}, "NetworkSettings": {"Bridge": "bridge", "ContainerID": "2ba11810c01d8ef3cdfb0d8427883a9136d813d70a1233d4e93c5ceee0d90e43", "IPAddress": "172.17.0.2", "IPPrefixLen": 16, "MacAddress": "0a:0a:0a:0a:0a:0a", "PortBindings": {}, "Ports": {}, "Links": {}, "Aliases": {}, "GlobalIPv6Address": null, "GlobalIPv6PrefixLen": null}}]
```

Tarea. Bind mount para compartir datos

Para poder superar esta actividad debes realizar las siguientes tareas:

1.- Crea una carpeta llamada saludo y dentro de ella crea un fichero llamado index.html con el siguiente contenido:

```
<h1>HOLA SOY XXXXXX</h1>
```

```
root@artacho-VirtualBox:/home/artacho# mkdir saludo
root@artacho-VirtualBox:/home/artacho# cd saludo/
root@artacho-VirtualBox:/home/artacho/saludo# echo "<h1>HOLA, SOY ANTONIO ARTACHO MONTERO</h1>" > index.html
root@artacho-VirtualBox:/home/artacho/saludo#
```

Deberás sustituir ese XXXXX por tu nombre.

2.- Una vez hecho esto arrancar dos contenedores basados en la imagen php:7.4- apache que hagan un bind mount de la carpeta saludo en la carpeta /var/www/html del contenedor.

3.- Uno de ellos deberá redireccionar su puerto 80 al 8181 y el otro al 8282. Y su nombres serán c1 y c2.

```
root@artacho-VirtualBox:/home/artacho/saludo# docker run --name c1 --mount type=bind,src=/home/artacho/saludo,dst=/var/www/html -d -p 8181:80 php:7.4-apache
f4a9a58fb60b88b74f99cb5eddf10cdc7acf9652a66dbb79115265f07db1ed9
root@artacho-VirtualBox:/home/artacho/saludo# docker run --name c2 --mount type=bind,src=/home/artacho/saludo,dst=/var/www/html -d -p 8282:80 php:7.4-apache
6f01e8d6443fb83dd5c0504dc19708f50ee81dea4ec54d4292029b65623bb57
root@artacho-VirtualBox:/home/artacho/saludo# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
6f01e8d6443f php:7.4-apache "docker-php-entrypoi..." 22 seconds ago Up 21 seconds 0.0.0.0:8282->80/tcp, :::8282->80/tcp c2
f4a9a58fb60b php:7.4-apache "docker-php-entrypoi..." About a minute ago Up About a minute 0.0.0.0:8181->80/tcp, :::8181->80/tcp c1
7190644c32d8 mariadb:latest "docker-entrypoint.s..." 3 days ago Exited (255) 2 hours ago 3306/tcp, 0.0.0.0:3336->80/tcp, :::3336->80/tcp bddd
7e4b56a23efa ubuntu:18.04 "bash" 3 days ago Up 58 minutes ubuntu
root@artacho-VirtualBox:/home/artacho/saludo#
```

4.- Para superar esta actividad deberéis entregar los siguientes pantallazos en un fichero comprimido:

Pantallazo con la orden correspondiente para arrancar el contenedor c1 (puerto 8181) realizando el bind mount solicitado. (pantallazo1.png/jpg).

```
root@artacho-VirtualBox:/home/artacho/saludo# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
6f01e8d6443f php:7.4-apache "docker-php-entrypoi..." 2 minutes ago Up 2 minutes 0.0.0.0 :8282->80/tcp, :::8282->80/tcp c2
f4a9a58fb60b php:7.4-apache "docker-php-entrypoi..." 3 minutes ago Up 3 minutes 0.0.0.0 :8181->80/tcp, :::8181->80/tcp c1
7e4b56a23efa ubuntu:18.04 "bash" 3 days ago Up About an hour ubuntu
root@artacho-VirtualBox:/home/artacho/saludo#
```

Pantallazo con la orden correspondiente para arrancar el contenedor c2 (puerto 8282) realizando el bind mount solicitado. (pantallazo2.png/jpg).

```
root@artacho-VirtualBox:/home/artacho/saludo# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
6f01e8d6443f php:7.4-apache "docker-php-entrypoi..." 2 minutes ago Up 2 minutes 0.0.0.0 :8282->80/tcp, :::8282->80/tcp c2
f4a9a58fb60b php:7.4-apache "docker-php-entrypoi..." 3 minutes ago Up 3 minutes 0.0.0.0 :8181->80/tcp, :::8181->80/tcp c1
7e4b56a23efa ubuntu:18.04 "bash" 3 days ago Up About an hour ubuntu
root@artacho-VirtualBox:/home/artacho/saludo#
```

Pantallazo donde se pueda apreciar que accediendo a la URL 127.0.0.1:8181 se puede ver el contenido de index.html (pantallazo3.png/jpg).

```
root@artacho-VirtualBox:/home/artacho/saludo
Archivo Editar Ver Buscar Terminal Ayuda
root@artacho-VirtualBox:/home/artacho/saludo# curl http://127.0.0.1:8181
<h1>HOLA, SOY ANTONIO ARTACHO MONTERO</h1>
root@artacho-VirtualBox:/home/artacho/saludo#
```

Pantallazo donde se pueda apreciar que accediendo a la URL 127.0.0.1:8282 se puede ver el contenido de index.html (pantallazo4.png/jpg).

```
root@artacho-VirtualBox:/home/artacho/saludo# curl http://127.0.0.1:8282
<h1>HOLA, SOY ANTONIO ARTACHO MONTERO</h1>
root@artacho-VirtualBox:/home/artacho/saludo#
```

Tarea. Creación de redes en docker.

En este Curso nos hemos centrado únicamente en las redes de Docker de tipo BRIDGE porque son las que más nos interesan como desarrolladores para montar entornos locales lo más similares posible a producción, sin tener que instalar nada o casi nada.

En esta actividad vamos a crear dos redes de ese tipo (BRIDGE) con los siguientes datos:

Red1

Nombre: red1

Dirección de red: 172.28.0.0

Máscara de red: 255.255.0.0

Gateway: 172.28.0.1

Red2

Nombre: red2

El resto de los datos será proporcionados automáticamente por Docker.

```
root@artacho-VirtualBox:/home/artacho/saludo# docker network create -d bridge -
root@artacho-VirtualBox:/home/artacho/saludo# docker network create -d bridge
--subnet 172.28.0.0/16 --gateway 172.28.0.1 red1
4a822925693193d8991018eb67139c9143f28c9884d20e8fb52b26fb719871d8
root@artacho-VirtualBox:/home/artacho/saludo# docker network create -d bridge red2
8dff3c9c8d0affcce09c7953592878881797ff21a1808ec337ce75b3dbf86e06
root@artacho-VirtualBox:/home/artacho/saludo#
```

Una vez creadas estas dos redes deberás entregar en un archivo comprimido los siguientes pantallazos:

1. Resultado de la orden docker network ls donde se puedan ver las dos redes creadas correctamente. (pantallazo1.jpg/png)

```
root@artacho-VirtualBox:/home/artacho/saludo# docker network ls
NETWORK ID      NAME      DRIVER      SCOPE
85bce1a555e3    bridge    bridge      local
be9e8136bd0a    host      host       local
ed9be0764437    none      null       local
4a8229256931    red1     bridge      local
8dff3c9c8d0a    red2     bridge      local
root@artacho-VirtualBox:/home/artacho/saludo#
```

2. Resultado de la orden docker network inspect red1 donde se pueda ver que la configuración de la red ha sido la correcta. (pantallazo2.jpg/png)

```
root@artacho-VirtualBox:/home/artacho/saludo# docker network inspect red1
[{"Name": "red1",
 "Id": "4a822925693193d8991018eb67139c9143f28c9884d20e8fb52b26fb719871d8",
 "Created": "2022-02-09T19:44:55.063911317+01:00",
 "Scope": "local",
 "Driver": "bridge",
 "EnableIPv6": false,
 "IPAM": {
     "Driver": "default",
     "Options": {},
     "Config": [
         {
             "Subnet": "172.28.0.0/16",
             "Gateway": "172.28.0.1"
         }
     ]
 },
 "Internal": false,
 "Attachable": false,
 "Ingress": false,
 "ConfigFrom": {
     "Network": ""
 },
 "ConfigOnly": false,
 "Containers": {},
 "Options": {},
 "Labels": {}
}]
```

3. Resultado de la orden docker network inspect red2 donde se pueda ver la configuración que Docker ha dado a la red. (pantallazo3.jpg/png)

```
root@artacho-VirtualBox:/home/artacho/saludo# docker network inspect red2
[
  {
    "Name": "red2",
    "Id": "8dff3c9c8d0affcce09c7953592878881797ff21a1808ec337ce75b3dbf86e06",
    "Created": "2022-02-09T19:45:17.012547267+01:00",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": {},
      "Config": [
        {
          "Subnet": "172.18.0.0/16",
          "Gateway": "172.18.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {},
    "Options": {},
    "Labels": {}
  }
]
```

Tarea. Conexión de redes a contenedores.

Una vez hemos creado las dos redes en el apartado anterior en esta actividad vamos a conectarlas a contenedores en ejecución.

Deberemos realizar los siguiente pasos:

1. Poner en ejecución un contenedor de la imagen ubuntu:20.04 que tenga como hostname host1, como IP 172.28.0.10 y que esté conectado a la red1. Lo llamaremos u1.

```
root@host1:/ 
Archivo Editar Ver Buscar Terminal Ayuda
root@artacho-VirtualBox:/home/artacho/saludo# docker run -it --name u1 --network red1 --ip 172.28.0.10 -h host1 ubuntu:20.04
Unable to find image 'ubuntu:20.04' locally
20.04: Pulling from library/ubuntu
08c01a0ec47e: Already exists
Digest: sha256:669e010b58baf5beb2836b253c1fd5768333f0d1dbc834f7c07
a4dc93f474be
Status: Downloaded newer image for ubuntu:20.04
root@host1:/#
```

2. Entrar en ese contenedor e instalar la aplicación ping (apt update && apt install inetutils-ping).

```
root@host1:/# apt update && apt install net-tools && apt install inetutils-ping
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [1542 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [177 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [11.3 MB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [839 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [25.8 kB]
Get:10 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [960 kB]
Get:11 http://archive.ubuntu.com/ubuntu focal/main amd64 Packages [1275 kB]
Get:12 http://archive.ubuntu.com/ubuntu focal/restricted amd64 Packages [33.4 kB]
Get:13 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [1027 kB]
Get:14 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1967 kB]
Get:15 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [29.4 kB]
Get:16 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1132 kB]
Get:17 http://archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [24.8 kB]
Get:18 http://archive.ubuntu.com/ubuntu focal-backports/main amd64 Packages [50.8 kB]
Fetched 21.0 MB in 25s (839 kB/s)
```

3. Poner en ejecución un contenedor de la imagen ubuntu:20.04 que tenga como hostname host2 y que esté conectado a la red2. En este caso será docker el que le de una IP correspondiente a esa red. Lo llamaremos u2.

```
root@artacho-VirtualBox:/home/artacho/saludo# docker run -it --name u2 --network red1 --ip 172.28.0
.10 -h host2 ubuntu:20.04
root@host2:/# cat /etc/hostname
host2
root@host2:/#
```

4. Entrar en ese contenedor e instalar la aplicación ping (apt update && apt install inetutils-ping).

```
root@host2:/# apt update && apt install net-tools && apt install inetutils-ping
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [960 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [11.3 MB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [25.8 kB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [839 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [1542 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [177 kB]
Get:11 http://archive.ubuntu.com/ubuntu focal/restricted amd64 Packages [33.4 kB]
Get:12 http://archive.ubuntu.com/ubuntu focal/main amd64 Packages [1275 kB]
Get:13 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [29.4 kB]
Get:14 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1967 kB]
Get:15 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1132 kB]
Get:16 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [1027 kB]
Get:17 http://archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [24.8 kB]
Get:18 http://archive.ubuntu.com/ubuntu focal-backports/main amd64 Packages [50.8 kB]
Fetched 21.0 MB in 47s (443 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
9 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
```

Una vez preparados estos contenedores debemos capturar los siguientes pantallazos:
Pantallazo donde se vea la configuración de red del contenedor u1. (pantallazo1.jpg/png)

```
root@host2:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 172.28.0.10 netmask 255.255.0.0 broadcast 172.28.255.255
                ether 02:42:ac:1c:00:0a txqueuelen 0 (Ethernet)
                RX packets 2660 bytes 21495117 (21.4 MB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 2573 bytes 143776 (143.7 KB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
                loop txqueuelen 1000 (Local Loopback)
                RX packets 8 bytes 762 (762.0 B)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 8 bytes 762 (762.0 B)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@host2:/#
```

Pantallazo donde se vea la configuración de red del contenedor u2. (pantallazo2.jpg/png)

```
root@host2:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.18.0.2 netmask 255.255.0.0 broadcast 172.18.255.255
        ether 02:42:ac:12:00:02 txqueuelen 0 (Ethernet)
        RX packets 2612 bytes 21494780 (21.4 MB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 2352 bytes 131866 (131.8 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        loop txqueuelen 1000 (Local Loopback)
        RX packets 8 bytes 762 (762.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 8 bytes 762 (762.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Pantallazo donde desde cualquiera de los dos contenedores se pueda ver que no podemos hacer ping al otro ni por ip ni por nombre. (pantallazo3.jpg/png)

```
root@artacho-VirtualBox:/home/artacho/saludo# docker exec -it u1 /bin/bash
root@host1:/# ping 172.18.0.2
PING 172.18.0.2 (172.18.0.2): 56 data bytes
^C--- 172.18.0.2 ping statistics ---
13 packets transmitted, 0 packets received, 100% packet loss
root@host1:/#
```

5. Una vez hemos constatado que los dos contenedores están en redes diferentes y aisladas, conectar la red2 al contenedor u1 mediante la orden docker network connect. Comprobar esta última conexión y capturar un último pantallazo (pantallazo4.jpg/png) donde se pueda comprobar que ahora, desde el contenedor u1, tenemos acceso al contenedor u2 mediante ping, tanto por nombre como por ip.

```
root@artacho-VirtualBox:/home/artacho/saludo# docker network connect red2 u1
root@artacho-VirtualBox:/home/artacho/saludo# docker start u1
u1
root@artacho-VirtualBox:/home/artacho/saludo# docker attach u1
root@host1:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.28.0.10 netmask 255.255.0.0 broadcast 172.28.255.255
        ether 02:42:ac:1c:00:0a txqueuelen 0 (Ethernet)
            RX packets 36 bytes 4080 (4.0 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 14 bytes 1316 (1.3 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.18.0.2 netmask 255.255.0.0 broadcast 172.18.255.255
        ether 02:42:ac:12:00:02 txqueuelen 0 (Ethernet)
            RX packets 23 bytes 2959 (2.9 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 0 bytes 0 (0.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        loop txqueuelen 1000 (Local Loopback)
            RX packets 0 bytes 0 (0.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 0 bytes 0 (0.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

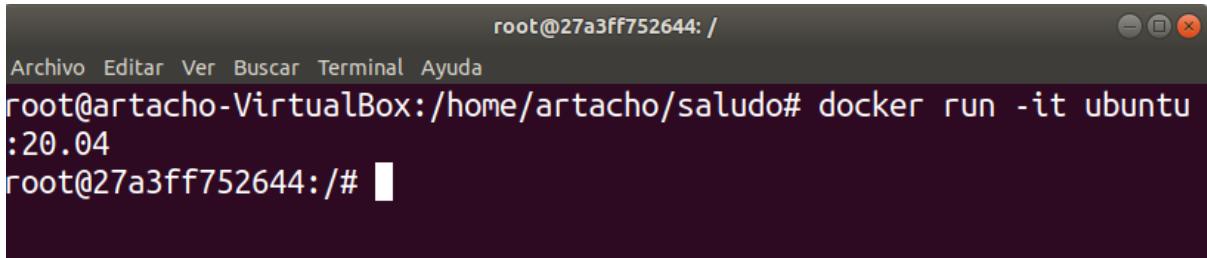
root@host1:/#
```

```
root@host1:/# ping 172.18.0.2
PING 172.18.0.2 (172.18.0.2): 56 data bytes
64 bytes from 172.18.0.2: icmp_seq=0 ttl=64 time=0.038 ms
64 bytes from 172.18.0.2: icmp_seq=1 ttl=64 time=0.043 ms
64 bytes from 172.18.0.2: icmp_seq=2 ttl=64 time=0.043 ms
64 bytes from 172.18.0.2: icmp_seq=3 ttl=64 time=0.042 ms
64 bytes from 172.18.0.2: icmp_seq=4 ttl=64 time=0.050 ms
64 bytes from 172.18.0.2: icmp_seq=5 ttl=64 time=0.047 ms
^C--- 172.18.0.2 ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.038/0.044/0.050/0.000 ms
root@host1:/#
```

Tarea. Creación de imágenes a partir de contenedores en ejecución.

Para poder superar esta actividad debes realizar las siguientes tareas:

- 1.- Arrancar un contenedor sobre la imagen ubuntu:20.04 y sobre él realizar las siguientes operaciones:



```
root@27a3ff752644: / 
Archivo Editar Ver Buscar Terminal Ayuda
root@artacho-VirtualBox:/home/artacho/saludo# docker run -it ubuntu
:20.04
root@27a3ff752644:/#
```

Instalación del editor nano (apt install nano).

```
root@27a3ff752644:/# apt install nano
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  hunspell
The following NEW packages will be installed:
  nano
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 269 kB of archives.
After this operation, 868 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal/main amd64 nano amd64 4.8-1ubuntu1 [269 kB]
Fetched 269 kB in 1s (425 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package nano.
(Reading database ... 4127 files and directories currently installed.)
Preparing to unpack .../nano_4.8-1ubuntu1_amd64.deb ...
Unpacking nano (4.8-1ubuntu1) ...
```

Instalación del editor vim (apt install vim).

```
root@27a3ff752644:/# apt install vim
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  alsa-topology-conf alsa-ucm-conf file libasound2 libasound2-data libcanberra0 libexpat1 libgpm2
  libltdl7 libmagic-mgc libmagic1 libmpdec2 libogg0 libpython3.8 libpython3.8-minimal
  libpython3.8-stdlib libreadline8 libssqlite3-0 libssl1.1 libtdb1 libvorbis0a libvorbisfile3
  mime-support readline-common sound-theme-freedesktop vim-common vim-runtime xxd xz-utils
Suggested packages:
  libasound2-plugins alsa-utils libcanberra-gtk0 libcanberra-pulse gpm readline-doc ctags vim-doc
  vim-scripts
The following NEW packages will be installed:
  alsa-topology-conf alsa-ucm-conf file libasound2 libasound2-data libcanberra0 libexpat1 libgpm2
  libltdl7 libmagic-mgc libmagic1 libmpdec2 libogg0 libpython3.8 libpython3.8-minimal
  libpython3.8-stdlib libreadline8 libssqlite3-0 libssl1.1 libtdb1 libvorbis0a libvorbisfile3
  mime-support readline-common sound-theme-freedesktop vim vim-common vim-runtime xxd xz-utils
0 upgraded, 30 newly installed, 0 to remove and 0 not upgraded.
Need to get 14.9 MB of archives.
After this operation, 70.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu focal/main amd64 libmagic-mgc amd64 1:5.38-4 [218 kB]
```

Instalación de las herramientas de red (apt install inetutils-tools).

```
root@27a3ff752644:/# apt install inetutils-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  inetutils-tools
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 74.6 kB of archives.
After this operation, 506 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 inetutils-tools amd64 2:1.9.4-1
1ubuntu0.1 [74.6 kB]
Fetched 74.6 kB in 0s (271 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package inetutils-tools.
(Reading database ... 7300 files and directories currently installed.)
Preparing to unpack .../inetutils-tools_2%3a1.9.4-11ubuntu0.1_amd64.deb ...
Unpacking inetutils-tools (2:1.9.4-11ubuntu0.1) ...
Setting up inetutils-tools (2:1.9.4-11ubuntu0.1) ...
root@27a3ff752644:/#
```

Instalación de las herramientas dns (apt install dnsutils).

```
root@27a3ff752644:/# apt install dnsutils
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bind9-dnsutils bind9-host bind9-libs krb5-locales libbsd0 libedit2 libgssapi-krb5-2 libicu66
  libjson-c4 libk5crypto3 libkeyutils1 libkrb5-3 libkrb5support0 liblmbdb0 libmaxminddb0 libuv1
  libxml2 tzdata
Suggested packages:
  krb5-doc krb5-user mmdb-bin
The following NEW packages will be installed:
  bind9-dnsutils bind9-host bind9-libs dnsutils krb5-locales libbsd0 libedit2 libgssapi-krb5-2
  libicu66 libjson-c4 libk5crypto3 libkeyutils1 libkrb5-3 libkrb5support0 liblmbdb0 libmaxminddb0
  libuv1 libxml2 tzdata
0 upgraded, 19 newly installed, 0 to remove and 0 not upgraded.
Need to get 11.6 MB of archives.
After this operation, 46.9 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu focal/main amd64 libbsd0 amd64 0.10.0-1 [45.4 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 tzdata all 2021e-0ubuntu0.20.04 [29
5 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libicu66 amd64 66.1-2ubuntu2.1 [851
5 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libjson-c4 amd64 0.13.1+dfsg-7ubunt
u0.3 [29.3 kB]
```

Creación del usuario usuario con contraseña usuario (adduser usuario)

```
root@27a3ff752644:/# adduser usuario
Adding user `usuario' ...
Adding new group `usuario' (1000) ...
Adding new user `usuario' (1000) with group `usuario' ...
Creating home directory `/home/usuario' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for usuario
Enter the new value, or press ENTER for the default
      Full Name []: usuario
      Room Number []:
      Work Phone []:
      Home Phone []:
      Other []:
Is the information correct? [Y/n] y
root@27a3ff752644:/#
```

2.- Tras realizar dichas instalaciones y utilizando la orden docker commit crear una imagen que se llame de la siguiente manera: TuNombreUsuarioDockerHub/a61 y subirla a DockerHub utilizando la orden docker push. Recordad que antes tendréis que hacer docker login.

```
root@artacho-VirtualBox:/home/artacho/saludo# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              NAMES
be4cf296f952        ubuntu:18.04       "bash"                 9 seconds ago     Exited (0) 8 seconds ago
go
27a3ff752644        ubuntu:20.04       "bash"                 18 minutes ago   Exited (127) 3 minutes
ago
45c64d97e233        ubuntu:20.04       "bash"                 28 minutes ago   Exited (0) 25 minutes
ago
996d2c028902        ubuntu:20.04       "bash"                 51 minutes ago   Exited (0) 18 minutes
ago
6f01e8d6443f        php:7.4-apache    "docker-php-entrypoi..."  About an hour ago  Up     About an hour
          0.0.0.0:8282->80/tcp, :::8282->80/tcp
f4a9a58fb60b        php:7.4-apache    "docker-php-entrypoi..."  About an hour ago  Up     About an hour
          0.0.0.0:8181->80/tcp, :::8181->80/tcp
7190644c32d8        mariadb:latest    "docker-entrypoint.s..."  3 days ago        Exited (255) 3 hours ago
go
          3306/tcp, 0.0.0.0:3336->80/tcp, :::3336->80/tcp
7e4b56a23efa        ubuntu:18.04       "bash"                 3 days ago        Up 2 hours
ubuntu
root@artacho-VirtualBox:/home/artacho/saludo# docker commit artacho antonioartacho/a61
sha256:bdd4bef947b0143f255a6e2da94536edb20075bcdce2e572092c0cd329e07497
```

3.- Deberéis entregar los siguientes pantallazos en un fichero comprimido:

Pantallazo de la orden para el arranque del contenedor (pantallazo1.png/jpg)

```
root@artacho-VirtualBox:/home/artacho/saludo# docker start artacho
artacho
root@artacho-VirtualBox:/home/artacho/saludo# docker ps
CONTAINER ID   IMAGE      COMMAND   CREATED          STATUS          PORTS
TS            ubuntu:18.04 "bash"    23 seconds ago   Up 1 second
```

Pantallazo de la orden docker commit para la creación de la nueva imagen

(pantallazo2.png/jpg)

```
root@artacho-VirtualBox:/home/artacho/saludo# docker commit artacho antonioartacho/a61
sha256:584e16694d1fc843c30c016186ca65ae3a31e390f79c81a33b204cd2d6b214f3
root@artacho-VirtualBox:/home/artacho/saludo# docker images
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
antonioartacho/a61  latest   584e16694d1f  10 seconds ago  63.2MB
mariadb              latest   45a5a43e143a  7 days ago    410MB
ubuntu               20.04   54c9d81cbb44  7 days ago    72.8MB
ubuntu               18.04   dcf4d4bef137  7 days ago    63.2MB
mysql                5.7     0712d5dc1b14  13 days ago   448MB
php                  7.3-apache 8f89e6e66d9b  2 weeks ago   465MB
php                  7.4-apache 4d3d9fe4d89c  2 weeks ago   469MB
httpd                latest   a8ea074f4566  2 weeks ago   144MB
debian               9        28c1b600f6da  2 weeks ago   101MB
jenkins/jenkins      lts     9aeee0d53624f  4 weeks ago   441MB
centos               8        5d0da3dc9764  4 months ago  231MB
tomcat               9.0.39   2703bbe9e9d4  15 months ago  648MB
tomcat               9.0.39-jdk11 2703bbe9e9d4  15 months ago  648MB
```

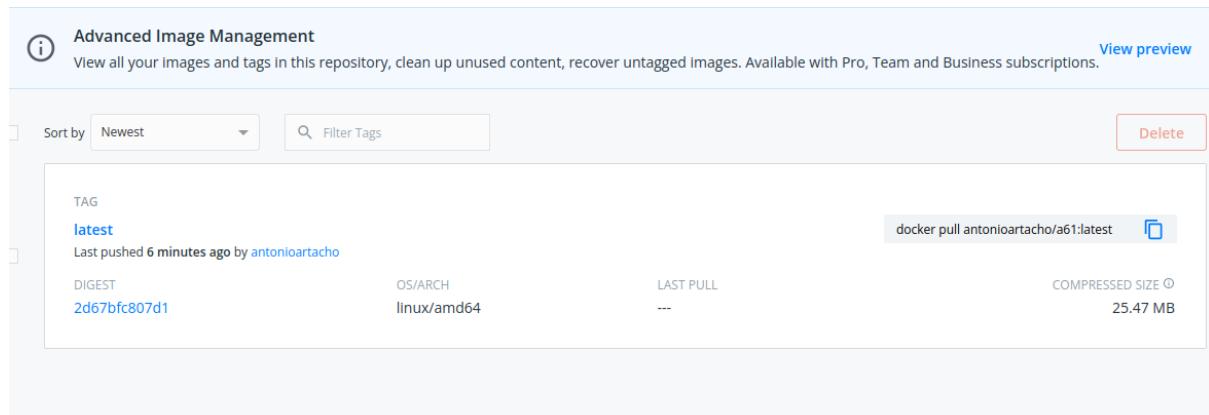
Pantallazo de la salida de la orden docker images donde se pueda ver que la imagen se ha creado (pantallazo3.png/jpg)

Pantallazo de la ejecución de la orden docker push (pantallazo4.png/jpg)

```
root@artacho-VirtualBox:/home/artacho/saludo# docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: antonioartacho
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@artacho-VirtualBox:/home/artacho/saludo# docker push antonioartacho/a61
Using default tag: latest
The push refers to repository [docker.io/antonioartacho/a61]
1dc52a6b4de8: Mounted from library/ubuntu
latest: digest: sha256:2d67bfc807d127969dc0f07fb221f0f9e1bde30c82f34519d324c73434bc5dcf size: 529
root@artacho-VirtualBox:/home/artacho/saludo#
```

Pantallazo de tu página de dockerHub donde se pueda ver que se ha creado esa imagen/repositorio (pantallazo5.png/jpg)



Tarea. Creación de imágenes a partir de ficheros

Dockerfile.

Para poder superar esta actividad debe realizar las siguientes tareas:

1.- Partiendo de la imagen php:7.4-apache construir un Dockerfile que realizar lo siguiente:

Instalar nano (apt install -y nano)

Instalar git (apt install -y git)

Colocar en el directorio raíz del servidor apache

(/var/www/html) dos ficheros:

index.html que contenga HOLA SOY XXXXXX

sustituyendo XXXXX por tu nombre

info.php que contiene el siguiente código <?php phpinfo(); ?>

2.- Una vez creado dicho Dockerfile construir la imagen, que se deberá llamar TuNombreUsuarioDockerHub/a62.

3.- Deberán entregar los siguientes pantallazos en un fichero comprimido:

Pantallazo del fichero Dockerfile (pantallazo1.png/jpg)

```
root@artacho-VirtualBox:/home/artacho
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.9.3 Dockerfile

FROM php:7.4-apache
RUN apt update && apt install -y nano && apt install -y git
WORKDIR /var/www/html
RUN echo "HOLA, SOY ANTONIO ARTACHO" > index.html
RUN echo "<?php phpinfo(); ?>" > info.php
```

Pantallazo de la ejecución de la orden docker build para la creación de la imagen a partir del fichero Dockerfile
(pantallazo2.png/jpg)

```
root@artacho-VirtualBox:/home/artacho# docker build -t antonioartacho/a61 .
Sending build context to Docker daemon 145.1MB
Step 1/5 : FROM php:7.4-apache
--> 4d3d9fe4d89c
Step 2/5 : RUN apt update && apt install -y nano && apt install -y git
--> Running in cf5bd863fce8
--> 91e0c684cef3
Step 4/5 : RUN echo "HOLA, SOY ANTONIO ARTACHO" > index.html
--> Running in bafa7c4b353a
Removing intermediate container bafa7c4b353a
--> 24bc708468b9
Step 5/5 : RUN echo "<?php phpinfo(); ?>" > info.php
--> Running in 590e1fc510dc
Removing intermediate container 590e1fc510dc
--> e0f8d22cd814
Successfully built e0f8d22cd814
Successfully tagged antonioartacho/a61:latest
root@artacho-VirtualBox:/home/artacho#
```

Pantallazo de la salida de la orden docker images donde se pueda ver que la imagen se ha creado (pantallazo3.png/jpg)

```
ubuntu
root@artacho-VirtualBox:/home/artacho# docker images
REPOSITORY          TAG           IMAGE ID      CREATED       SIZE
antonioartacho/a61  latest        e0f8d22cd814  2 minutes ago  521MB
```

Pantallazo de la ejecución de la orden docker push
(pantallazo4.png/jpg)

```
root@artacho-VirtualBox:/home/artacho# docker push antonioartacho/a61
Using default tag: latest
The push refers to repository [docker.io/antonioartacho/a61]
cf2dc647930c: Pushed
07150efd0fd0: Pushed
c684f4fb79c9: Pushing 44.41MB/52.27MB
0aef70e869ab: Mounted from library/php
643fb68a65ee: Mounted from library/php
bf5746c58444: Mounted from library/php
c610dfa76ee3: Waiting
ad19df8301b0: Waiting
67f81209206e: Waiting
03c9b818419c: Waiting
e920b2fd0dec: Waiting
c154893b7027: Waiting
994a8a93cc06: Waiting
45a669946c79: Waiting
ebc3f399f9c7: Waiting
7d0ebbe3f5d2: Waiting
```

Pantallazo de tu página de dockerHub donde se pueda ver que se ha creado esa imagen/repositorio (pantallazo5.png/jpg)

The screenshot shows the Docker Hub interface for the repository 'antonioartacho/a61'. The 'Tags' tab is active, displaying a single tag named 'latest'. Below the tag, it says 'Last pushed a few seconds ago by antonioartacho'. To the right, there's a 'docker pull' command and a delete button. At the bottom, it shows the digest '879d02fad7c3', the OS/ARCH 'linux/amd64', the last pull date '---', and the compressed size '187.58 MB'.

Cometí el error de llamar a la nueva imagen igual que a la anterior, sin embargo, dado que las versiones que se usan son distintas, DockerHub las acepta sin problema.

Módulo 7.- Compartir docker compose

Lo primero que debemos hacer es crear el fichero docker-compose.yaml, el cual tendrá el siguiente aspecto:

```
root@usuario-HP-EliteDesk-800-G1-SFF: /home/usuario
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.9.3                               docker-compose.yaml

Version: '3'
services:
  #-----#
  # SERVICIO SERVIDOR WEB (php:7.4-apache con mysqli y código WP descargado
  #-----#
  web:
    # IMAGEN USADA
    image: jperjim398/miwp
    # NOMBRE QUE LE VOY A DAR AL CONTENEDOR
    container_name: web
    # REDIRECCIÓN DE PUERTOS
    ports:
      - 8181:80
    # SERVICIOS QUE TIENEN QUE ARRANCAR ANTES DE ARRANCAR ESTE
    depends_on:
      - datos
    # REDES A USAR
    networks:
      - ejemplo

  #-----#
  # SERVICIO SERVIDOR DE BASE DE DATOS MARIADB
  #-----#


  # NOMBRE DEL SERVICIO
  datos:
    # IMAGEN USADA
    image: mariadb
    # NOMBRE QUE LE VOY A DAR AL CONTENEDOR
```

```
# NOMBRE DEL SERVICIO
datos:
  # IMAGEN USADA
  image: mariadb
  # NOMBRE QUE LE VOY A DAR AL CONTENEDOR
  container_name: bd
  # LISTA DE VALORES DE ENTORNO CON SUS VALORES
  environment:
    MYSQL_ROOT_PASSWORD: 123456
    MYSQL_DATABASE: wordpress
    MYSQL_USER: antonio_artacho
    MYSQL_PASSWORD: antonio_artacho
  # REDIRECCIÓN DE PUERTOS
  ports:
    - 3316:3306
  # VOLUMENES A USAR POR EL CONTENEDOR
  volumes:
    # DOCKER VOLUME
```

A continuación introducimos el comando “docker-compose” up en el directorio en el que se encuentra el archivo:

```
root@usuario-HP-EliteDesk-800-G1-SFF:/home/usuario# nano docker-compose.yml
root@usuario-HP-EliteDesk-800-G1-SFF:/home/usuario# docker-compose up
Creating bd ... done
Creating web ... done
Attaching to bd, web
bd    | 2022-02-10 07:51:52+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:10.6.4+maria~focal started.
bd    | 2022-02-10 07:51:52+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
bd    | 2022-02-10 07:51:52+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:10.6.4+maria~focal started.
bd    | 2022-02-10 07:51:52+00:00 [Note] [Entrypoint]: Initializing database files
web  | AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.19.0.3. Set the 'ServerName' directive globally to suppress this message
web  | AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.19.0.3. Set the 'ServerName' directive globally to suppress this message
web  | [Thu Feb 10 07:51:52.957447 2022] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.51 (Debian) PHP/7.4.27 configured -- resuming normal operations
web  | [Thu Feb 10 07:51:52.957486 2022] [core:notice] [pid 1] AH00094: Command line: 'apache2 -D FOREGROUND'
bd    | 2022-02-10 7:51:52 0 [Warning] You need to use --log-bin to make --expire-logs-days or --binlog-expire-logs-seconds work.
bd
bd
bd  | PLEASE REMEMBER TO SET A PASSWORD FOR THE MariaDB root USER !
bd  | To do so, start the server, then issue the following command:
bd  | '/usr/bin/mysql_secure_installation'
bd
bd  | which will also give you the option of removing the test
bd  | databases and anonymous user created by default. This is
bd  | strongly recommended for production servers.
bd
bd  | See the MariaDB Knowledgebase at https://mariadb.com/kb or the
bd  | MySQL manual for more instructions.
bd
bd  | Please report any problems at https://mariadb.org/jira
bd
```

Por último escribimos el comando “docker ps -a” o “docker inspect web/bd” para comprobar que los contenedores definidos en el archivo docker-compose.yml se han creado correctamente.

```
root@usuario-HP-EliteDesk-800-G1-SFF:/home/usuario# docker ps -a
CONTAINER ID   IMAGE          COMMAND       CREATED        STATUS          PORTS     NAMES
08af120f80fe   php:7.4-apache "docker-php-entrypoi..."   About a minute ago   Exited (0) 6 seconds ago
9ccc6b43dd7b   mariadb      "docker-entrypoint.s..."   About a minute ago   Exited (0) 5 seconds ago
root@usuario-HP-EliteDesk-800-G1-SFF:/home/usuario# docker inspect web
[
  {
    "Id": "08af120f80fe1ec4bca6567a00e8c0fcc145c42dee4aee6c7eb4582372a92a8e",
    "Created": "2022-02-10T07:51:52.262082371Z",
    "Path": "docker-php-entrypoint",
    "Args": [
      "apache2-foreground"
    ],
    "State": {
      "Status": "exited",
      "Running": false,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 0,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2022-02-10T07:51:52.766104197Z",
      "FinishedAt": "2022-02-10T07:53:39.847556648Z"
    },
    "NetworkSettings": {
      "Bridge": "bridge",
      "ContainerID": "08af120f80fe1ec4bca6567a00e8c0fcc145c42dee4aee6c7eb4582372a92a8e",
      "EndpointID": "08af120f80fe1ec4bca6567a00e8c0fcc145c42dee4aee6c7eb4582372a92a8e",
      "Gateway": "172.19.0.1",
      "GlobalIPv6Address": null,
      "GlobalIPv6PrefixLen": null,
      "IPAddress": "172.19.0.3",
      "IPPrefixLen": 16,
      "LinkLocalIPv6Address": null,
      "LinkLocalIPv6PrefixLen": null,
      "MacAddress": "00:0e:8c:0f:cc:14",
      "NetworkMode": "bridge",
      "Ports": {},
      "SecondaryIPAddresses": [],
      "SecondaryIPv6Addresses": [],
      "StartIP": null
    }
  }
]
```

Módulo 8.- Asegurando contendores

Usamos el comando docker trust para comprobar que todo esté correcto en mi repositorio.

```
root@usuario-HP-EliteDesk-800-G1-SFF:/home/usuario# docker trust inspect
antonioartacho/a61
[]
No signatures or cannot access antonioartacho/a61
root@usuario-HP-EliteDesk-800-G1-SFF:/home/usuario#
```

En el archivo /home/usuario/.bashrc añadiremos la siguiente línea al final.

```
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.9.3                               /home/usuario/.bashrc                                Modificado
Archivos
alias alert='notify-send --urgency=low -i "$( [ $? = 0 ] && echo terminal || echo error)" "$(history|tail -n1|sed -e '\$' -e '\''s/[0-9]\{1,\}\. [0-9]\{1,\} [0-9]\{1,\} [0-9]\{1,\}/>\1\2\3\4/g')"'
```

```
# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi
export DOCKER_CONTENT_TRUST=1
```

A continuación, generamos la llave **micontra**.

```
root@usuario-HP-EliteDesk-800-G1-SFF:/home/usuario# docker trust key generate micontra
Generating key for micontra...
Enter passphrase for new micontra key with ID bf4b32e:
Repeat passphrase for new micontra key with ID bf4b32e:
Successfully generated and loaded private key. Corresponding public key available: /home/usuario/micontra.pub
root@usuario-HP-EliteDesk-800-G1-SFF:/home/usuario#
```

```
root@usuario-HP-EliteDesk-800-G1-SFF:/home/usuario# docker trust signer add --key micontra.pub artacho antonioartacho/a61
Adding signer "artacho" to antonioartacho/a61...
Initializing signed repository for antonioartacho/a61...
Enter passphrase for root key with ID e84161a:
Enter passphrase for new repository key with ID 77e61cf:
Repeat passphrase for new repository key with ID 77e61cf:
Passphrases do not match. Please retry.
Enter passphrase for new repository key with ID 77e61cf:
Repeat passphrase for new repository key with ID 77e61cf:
Successfully initialized "antonioartacho/a61"
Successfully added signer: artacho to antonioartacho/a61
```

Para finalizar descargamos la imagen especificada anteriormente. Es recomendable hacer una inspección de la imagen.

```
root@usuario-HP-EliteDesk-800-G1-SFF:/home/usuario# docker trust sign antonioartacho/a61:latest
Signing and pushing trust data for local image antonioartacho/a61:latest, may overwrite remote trust data
The push refers to repository [docker.io/antonioartacho/a61]
cf2dc647930c: Layer already exists
07150efd0fd0: Layer already exists
c684f4fb79c9: Layer already exists
0aef70e869ab: Layer already exists
643fb68a65ee: Layer already exists
bf5746c58444: Layer already exists
c610dfa76ee3: Layer already exists
ad19df8301b0: Layer already exists
67f81209206e: Layer already exists
03c9b818419c: Layer already exists
e920b2fd0dec: Layer already exists
c154893b7027: Layer already exists
994a8a93cc06: Layer already exists
45a669946c79: Layer already exists
ebc3f399f9c7: Layer already exists
7d0ebbe3f5d2: Layer already exists
latest: digest: sha256:879d02fad7c304477d531aba493e0409127da70d2e43d906eb08731c816b84de size: 3661
Signing and pushing trust metadata
Enter passphrase for micontra key with ID bf4b32e:
Passphrase incorrect. Please retry.
Enter passphrase for micontra key with ID bf4b32e:
Passphrase incorrect. Please retry.
Enter passphrase for micontra key with ID bf4b32e:
Successfully signed docker.io/antonioartacho/a61:latest
root@usuario-HP-EliteDesk-800-G1-SFF:/home/usuario#
```

```
root@usuario-HP-EliteDesk-800-G1-SFF:/home/usuario# docker trust inspect antonioartacho/a61:latest
[
  {
    "Name": "antonioartacho/a61:latest",
    "SignedTags": [
      {
        "SignedTag": "latest",
        "Digest": "879d02fad7c304477d531aba493e0409127da70d2e43d906eb08731c816b84de",
        "Signers": [
          "artacho"
        ]
      }
    ],
    "Signers": [
      {
        "Name": "artacho",
        "Keys": [
          {
            "ID": "bf4b32e1af46ef27913f0cb7d412d1b8fc49ef345a7971aa8c1898cc6c3067a4"
          }
        ]
      }
    ],
    "AdministrativeKeys": [
      {
        "Name": "Root",
        "Key": "bf4b32e1af46ef27913f0cb7d412d1b8fc49ef345a7971aa8c1898cc6c3067a4"
      }
    ]
  }
]
```

Módulo 9.- Integración continua con Docker y Jenkins

Primero creamos la el contenedor con la imagen de jenkins.

```
root@usuario-HP-EliteDesk-800-G1-SFF:/home/usuario# docker run -d --name serverJenkins -p 9393:8080 -p 50001:50000 jenkins/jenkins:lts
2ad4ffc4a1f38e6f033b7cf48eb22a3edbbaf4d0c15e2155a35a17e4893ffd
root@usuario-HP-EliteDesk-800-G1-SFF:/home/usuario# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS
PORTS NAMES
2ad4ffc4a1f3 jenkins/jenkins:lts "/sbin/tini -- /usr/..." About a minute ago Up About a minute
0.0.0.0:9393->8080/tcp, :::9393->8080/tcp, 0.0.0.0:50001->50000/tcp, :::50001->50000/tcp
serverJenkins
```

Asignamos la configuración siguiente y accedemos al localhost correspondiente.

```
root@usuario-HP-EliteDesk-800-G1-SFF:/home/usuario# docker exec -u root serverJenkins more /var/jenkins_home/secrets/initialAdminPassword
::::::::::::::::::
/var/jenkins_home/secrets/initialAdminPassword
::::::::::::::::::
2a1170f7cf8e4953934cfbfd1e8ad000
root@usuario-HP-EliteDesk-800-G1-SFF:/home/usuario#
```

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

`/var/jenkins_home/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

.....

Continue

Getting Started

Create First Admin User

Usuario:

Contraseña:

Confirma la contraseña:

Nombre completo:

Dirección de email:

Jenkins 2.319.2 Skip and continue as admin Save and Continue

Ya tenemos cuenta en la página oficial de Jenkins.

The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with links: 'Nueva Tarea', 'Personas', 'Historial de trabajos', 'Administrador Jenkins' (which is selected and highlighted in blue), 'Mis vistas', 'Lockable Resources', 'New View', and 'Trabajos en la cola'. The main content area has a title 'Administrar Jenkins'. It displays a message: 'Hay una nueva versión de Jenkins disponible (2.319.3). [descargar \(listado de cambios\)](#)'. Below this, there's a yellow box with the text: 'Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#)'. It contains three buttons: 'Set up agent', 'Set up cloud', and 'Dismiss'. At the bottom, it says 'Advertencias publicadas para los siguientes componentes instalados:' followed by 'Jenkins 2.319.2 programa y librerías: [DoS vulnerability in bundled XStream library](#)'. A button 'Configurar las advertencias mostradas' is also present.

Activamos los plugins necesarios.

Actualizaciones disponibles		Todos los plugins	Plugins instalados	Configuracion avanzada
Install	Name		Version	Released
<input type="checkbox"/>	Javadoc	This plugin adds Javadoc support to Jenkins.	217.v905b_86277a_2a_	Hace 2 días 0 Hor
<input type="checkbox"/>	Authentication Tokens API	This plugin provides an API for converting credentials into authentication tokens in Jenkins.	1.4	Hace 1 Año 7 Mes
<input checked="" type="checkbox"/>	Maven Integration	Plugins relacionados con la forma de ejecutar trabajos This plug-in provides, for better and for worse, a deep integration of Jenkins and Maven: Automatic triggers between projects depending on SNAPSHOTs, automated configuration of various Jenkins publishers (Junit, ...).	3.16	Hace 1 Mes 20 días

```
root@usuario-HP-EliteDesk-800-G1-SFF:/home/usuario# docker run -d -it --name cep -p 9292:8080 -p 22 22:22 antonioartacho/a61
c944a4dfb22107c92c58c72fcefe59e6e9cdebf12983d31190382e1a344ef79b
root@usuario-HP-EliteDesk-800-G1-SFF:/home/usuario# docker ps
CONTAINER ID   IMAGE          COMMAND       CREATED      STATUS      PORTS          NAMES
S
c944a4dfb221   antonioartacho/a61   "docker-php-entrypoi..."   11 seconds ago   Up 10 seconds   80/tcp, 0.0.0.0:2222->22/tcp, 0.0.0.0:9292->8080/tcp, 0.0.0.0:9292->8080/tcp   cep
```

Creamos el contendor y comprobamos que podemos acceder a tomcat.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner *  Artacho98 ▾	Repository name * <input style="width: 150px; border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px;" type="text" value="dockerimage-from-jenkins-pipelin"/> ✓
Great repository names are short and memorable. Need inspiration? How about fluffy-telegram ?	
Description (optional) <div style="border: 1px solid #ccc; width: 100%; height: 40px; margin-top: 5px;"></div>	
<input checked="" type="radio"/>  Public Anyone on the internet can see this repository. You choose who can commit.	
<input type="radio"/>  Private You choose who can see and commit to this repository.	
<hr/>	
Username with password <div style="float: right; border: 1px solid #ccc; padding: 2px 5px; margin-bottom: 5px;">?</div>	
Scope <div style="float: right; border: 1px solid #ccc; padding: 2px 5px; margin-bottom: 5px;">?</div>	
<input type="text" value="Global (Jenkins, nodes, items, all child items, etc)"/> <div style="float: right; border: 1px solid #ccc; padding: 2px 5px; margin-bottom: 5px;">?</div>	
Username <div style="float: right; border: 1px solid #ccc; padding: 2px 5px; margin-bottom: 5px;">?</div>	
<input type="text" value="a_antartmon@iestriana.com"/> <div style="float: right; border: 1px solid #ccc; padding: 2px 5px; margin-bottom: 5px;">?</div>	
<input checked="" type="checkbox"/> Treat username as secret <div style="float: right; border: 1px solid #ccc; padding: 2px 5px; margin-bottom: 5px;">?</div>	
Password <div style="float: right; border: 1px solid #ccc; padding: 2px 5px; margin-bottom: 5px;">?</div>	
<input type="password" value="*****"/> <div style="float: right; border: 1px solid #ccc; padding: 2px 5px; margin-bottom: 5px;">?</div>	
ID <div style="float: right; border: 1px solid #ccc; padding: 2px 5px; margin-bottom: 5px;">?</div>	
<input type="text" value="gitprueba"/> <div style="float: right; border: 1px solid #ccc; padding: 2px 5px; margin-bottom: 5px;">?</div>	
Description <div style="float: right; border: 1px solid #ccc; padding: 2px 5px; margin-bottom: 5px;">?</div>	
<input type="text" value="pipeline para el curso"/> <div style="float: right; border: 1px solid #ccc; padding: 2px 5px; margin-bottom: 5px;">?</div>	
<input style="background-color: #0072BC; color: white; border: 1px solid #0072BC; padding: 5px; border-radius: 5px; font-weight: bold; margin-top: 10px;" type="button" value="OK"/>	

Por último construimos el Pipeline y nos aseguramos de que en el localhost nos muestra algún mensaje.

Bindings

Username and password (separated) ?

Username Variable ?

Password Variable ?

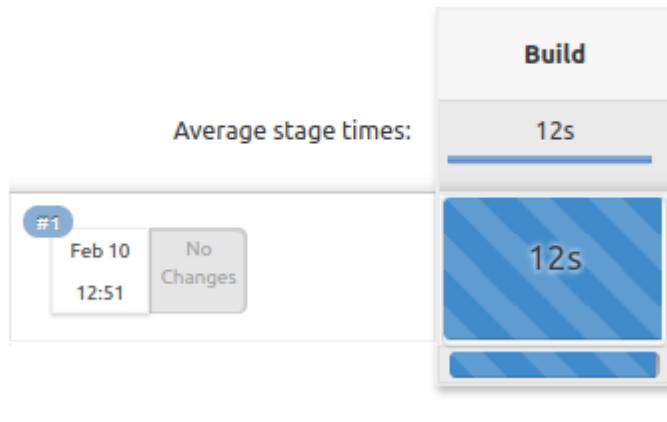
Credentials ? Add

Borrar

Añadir Generate Pipeline Script

```
withCredentials([usernamePassword(credentialsId: 'gitprueba', passwordVariable: 'password', usernameVariable: 'usuario')]) {
    // some block
}
```

Stage View



Comprobamos que carga el mensaje de bienvenida.

← → ⌂ 127.0.0.1:9292/webapp/

YouTube Colonist.io lichess Clases Drive MLB WhatsApp

Hello, World!