

TALLER DOCKER

SERGIO PERALTA

LAURA VALENTINA TELLEZ

JHONATHAN TORRES

PRESENTADO A:

WILSON CALVO

UNIVERSIDAD COOPERATIVA DE COLOMBIA

FACULTAD DE INGENIERIA

SEDE BOGOTÁ

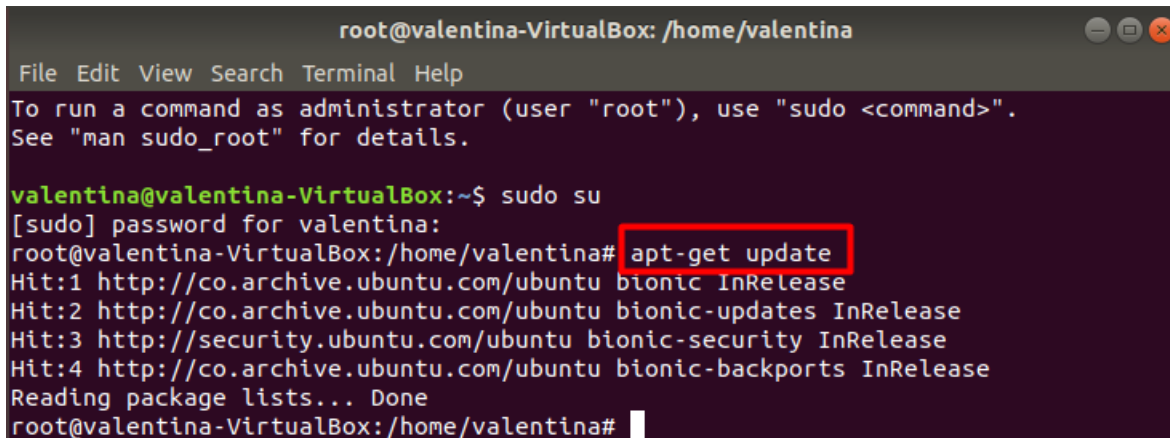
2023

Taller 1: Introducción a Docker

Paso 1: Instalar e Iniciar Docker

Ejecuta como root o con permisos de root el siguiente comando, para actualizar los repositorios en la máquina virtual recién creada:

apt-get update

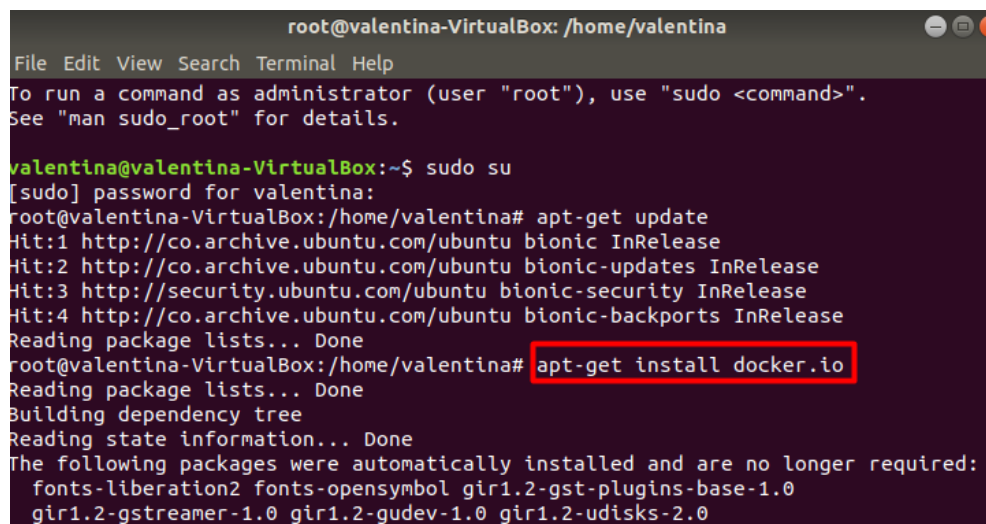
A terminal window titled 'root@valentina-VirtualBox: /home/valentina'. The prompt is 'valentina@valentina-VirtualBox:~\$'. The user enters 'sudo su'. The prompt changes to 'root@valentina-VirtualBox:/home/valentina#'. The user enters 'apt-get update', which is highlighted with a red box. The terminal shows four hits from Ubuntu repositories and the message 'Reading package lists... Done'.

```
root@valentina-VirtualBox: /home/valentina
File Edit View Search Terminal Help
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

valentina@valentina-VirtualBox:~$ sudo su
[sudo] password for valentina:
root@valentina-VirtualBox:/home/valentina# apt-get update
Hit:1 http://co.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://co.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:4 http://co.archive.ubuntu.com/ubuntu bionic-backports InRelease
Reading package lists... Done
root@valentina-VirtualBox:/home/valentina#
```

Luego ejecuta el siguiente comando para instalar Docker

apt-get install docker.io

A terminal window titled 'root@valentina-VirtualBox: /home/valentina'. The prompt is 'valentina@valentina-VirtualBox:~\$'. The user enters 'sudo su'. The prompt changes to 'root@valentina-VirtualBox:/home/valentina#'. The user enters 'apt-get update', which is highlighted with a red box. The terminal shows four hits from Ubuntu repositories and the message 'Reading package lists... Done'. The user then enters 'apt-get install docker.io', which is highlighted with a red box. The terminal shows the message 'Reading package lists... Done', 'Building dependency tree', 'Reading state information... Done', and a list of packages that were automatically installed and are no longer required.

```
root@valentina-VirtualBox: /home/valentina
File Edit View Search Terminal Help
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

valentina@valentina-VirtualBox:~$ sudo su
[sudo] password for valentina:
root@valentina-VirtualBox:/home/valentina# apt-get update
Hit:1 http://co.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://co.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:4 http://co.archive.ubuntu.com/ubuntu bionic-backports InRelease
Reading package lists... Done
root@valentina-VirtualBox:/home/valentina# apt-get install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
 fonts-liberation2 fonts-opensymbol gir1.2-gst-plugins-base-1.0
 gir1.2-gstreamer-1.0 gir1.2-gudev-1.0 gir1.2-udisks-2.0
```

Luego iniciar el servicio si esta apagado con:

/etc/init.d/docker start

```
Processing triggers for systemd (237-3ubuntu10.52) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
root@valentina-VirtualBox:/home/valentina# /etc/init.d/docker start
bash: /etc/init.d/docker: No such file or directory
root@valentina-VirtualBox:/home/valentina#
```

Paso 2: Bajar imágenes del repositorio de Docker en específico la de Ubuntu 18.04

Para bajar una imagen utilizamos el comando docker pull, en este caso bajaremos la imagen de Ubuntu 18.04 para esto ejecutamos el comando:

docker pull ubuntu:18.04

```
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
root@valentina-VirtualBox:/home/valentina# /etc/init.d/docker start
bash: /etc/init.d/docker: No such file or directory
root@valentina-VirtualBox:/home/valentina# docker pull ubuntu:18.04
18.04: Pulling from library/ubuntu
0c5227665c11: Pull complete
Digest: sha256:8aa9c2798215f99544d1ce7439ea9c3a6dfd82de607da1cec3a8a2fae005931b
Status: Downloaded newer image for ubuntu:18.04
docker.io/library/ubuntu:18.04
root@valentina-VirtualBox:/home/valentina#
```

Paso 3: Crear container con Docker y administrarlo

Utilizar el comando docker run, en este caso crearemos un container con Ubuntu 18.04 y ejecutaremos dentro el comando /bin/bash, el cual es una consola o terminal, nótese que al terminar la ejecución del /bin/bash con exit, el comando con el que fue creado el container termina su ejecución y el container es finalizado con el estado exit(0), finalizado más no eliminado

docker run -it ubuntu:18.04 /bin/bash

```
0c5227665c11: Pull complete
Digest: sha256:8aa9c2798215f99544d1ce7439ea9c3a6dfd82de607da1cec3a8a2fae005931b
Status: Downloaded newer image for ubuntu:18.04
docker.io/library/ubuntu:18.04
root@valentina-VirtualBox:/home/valentina# docker run -it ubuntu:18.04 /bin/bas
```

Al ser creado el container creara un id único, el cual puedes obtener cuando se coloca el nuevo prompt, al terminar de ejecutar el comando anterior, verás algo como:

root@e6b272bfcbc2:/#

```
root@197981c337ae:/#
```

Continuando, luego ejecuta dentro del container el comando `ls` para mostrar los archivos actuales en la carpeta que te encuentres

ls

```
root@valentina-VirtualBox:/home/valentina# docker run -it ubuntu:18.04 /bin/bash
root@197981c337ae:/# ls
bin  dev  home  lib64  mnt  proc  run  srv  tmp  var
boot  etc  lib  media  opt  root  sbin  sys  usr
```

En otra terminal ingresa y ejecuta el comando `docker ps -a` podrás ver que dicho container se encuentra en estado activo: verás una salida como la siguiente:

docker ps -a

```
root@valentina-VirtualBox:/home/valentina# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
2c4403c25a90   ubuntu:18.04   "/bin/bash"             About a minute ago    Exited (0) 57
seconds ago    frosty_ardinghelli
197981c337ae   ubuntu:18.04   "/bin/bash"             14 minutes ago       Exited (127) 3
minutes ago    crazy_kalam
```

Ahora para volver a iniciar el container ejecuta:

docker start id_container

```
root@valentina-VirtualBox:/home/valentina# docker start 72bb9dad1f39
72bb9dad1f39
root@valentina-VirtualBox:/home/valentina#
```

Esto reinicia la ejecución de container con el id que hallas colocado

Ahora para volver a ingresar, puedes ejecutar

docker exec -it id_container /bin/bash

```
72bb9dad1f39
root@valentina-VirtualBox:/home/valentina# docker exec -it 72bb9dad1f39 /bin/bash
sh
root@72bb9dad1f39:/#
```

Paso 4: Detener y borrar el container

Para detener el container puedes ejecutar el comando:

docker stop id_container

```
72bb9dad1f39
root@valentina-VirtualBox:/home/valentina# docker stop 72bb9dad1f39
72bb9dad1f39
root@valentina-VirtualBox:/home/valentina#
```

Esto solo detendrá su ejecución, pero no lo eliminará

Para poder eliminar y borrar el container debes de ejecutar el siguiente comando:

docker rm id_container

```
root@valentina-VirtualBox:/home/valentina# docker rm 72bb9dad1f39
72bb9dad1f39
root@valentina-VirtualBox:/home/valentina#
```

Para poder eliminarlo debes detener primero el container. En algunos casos debes usar la opción -f para forzar que se borre, según las dependencias con otras imágenes o containers. Puedes comprobar que ya no exista con el comando:

docker ps -a

```
root@valentina-VirtualBox:/home/valentina# docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
2c4403c25a90	ubuntu:18.04	"/bin/bash"	34 minutes ago	Exited (0) 34 minutes ago
197981c337ae	ubuntu:18.04	"/bin/bash"	48 minutes ago	Exited (127) 37 minutes ago

Taller 2: Acceso público de containers

Paso 1: Crear container y exponer los puertos

Luego de haber ingresado a tu máquina virtual con Docker instalado, ejecutar el siguiente comando para crear una máquina virtual con Ubuntu:18.04

docker run -it -d -p 8001:80 --name=apache ubuntu:18.04 /bin/bash

```
root@valentina-VirtualBox: /home/valentina
File Edit View Search Terminal Help
valentina@valentina-VirtualBox:~$ sudo su
[sudo] password for valentina:
root@valentina-VirtualBox:/home/valentina# docker run -it -d -p 8001:80 --name=
apache ubuntu:18.04 /bin/bash
67ff47f4c655161105735542da4b79a7997595ece176cfd66c8b520a147d6beb
root@valentina-VirtualBox:/home/valentina#
```

Paso 2: Acceder al container que está en segundo plano

Para acceder al container debes de ejecutar el comando:

docker exec -it apache /bin/bash

```
root@67ff47f4c655: /
File Edit View Search Terminal Help
valentina@valentina-VirtualBox:~$ sudo su
[sudo] password for valentina:
root@valentina-VirtualBox:/home/valentina# docker run -it -d -p 8001:80 --name=
apache ubuntu:18.04 /bin/bash
67ff47f4c655161105735542da4b79a7997595ece176cfd66c8b520a147d6beb
root@valentina-VirtualBox:/home/valentina# docker exec -it apache /bin/bash
root@67ff47f4c655:/#
```

Luego de ingresar al container actualiza el repositorio de ubuntu:

apt-get update

```
root@valentina-VirtualBox:/home/valentina# docker exec -it apache /bin/bash
root@67ff47f4c655:/# apt-get update
Get:1 http://archive.ubuntu.com/ubuntu bionic InRelease [242 kB]
Get:2 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [83.3 kB]
```

Luego instala Apache con el siguiente comando:

```
apt-get install apache2
```

```
root@67ff47f4c655: /
File Edit View Search Terminal Help
s [1606 kB]
Fetched 27.3 MB in 7s (4040 kB/s)
Reading package lists... Done
root@67ff47f4c655:/# apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils file libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libasn1-8-heimdal libexpat1
  libgdbm-compat4 libgdbm5 libgssapi3-heimdal libhcrypto4-heimdal
```

Luego inicia apache con el comando

```
/etc/init.d/apache2 start
```

```
Processing triggers for libc-bin (2.27-3ubuntu1.6) ...
root@67ff47f4c655:/# /etc/init.d/apache2 start
* Starting Apache httpd web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
*
root@67ff47f4c655:/#
```

Luego ejecuta exit para salir del container

exit

```
Enabling conf serve-cgi-bin.
Enabling site 000-default.
invoke-rc.d: could not determine current runlevel
invoke-rc.d: policy-rc.d denied execution of start.
Processing triggers for libc-bin (2.27-3ubuntu1.6) ...
root@67ff47f4c655:/# /etc/init.d/apache2 start
* Starting Apache httpd web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
*
root@67ff47f4c655:/# exit
exit
root@valentina-VirtualBox:/home/valentina#
```

Paso 3: Accede al sitio localizado enlazado al container

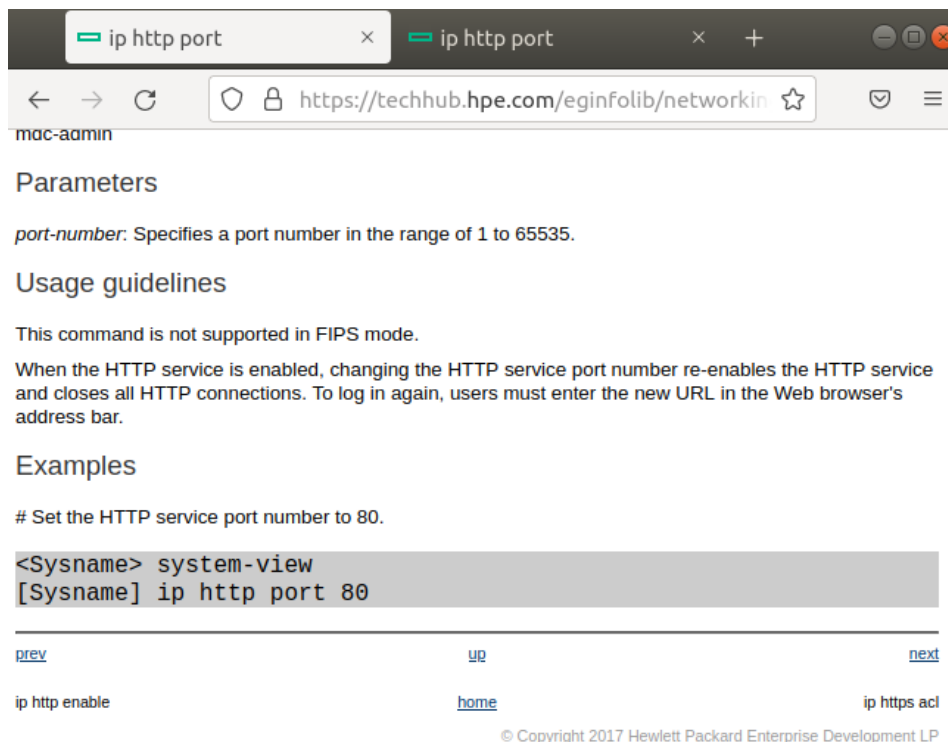
Para esto debes de obtener el ip de tu máquina virtual, en este caso el IP público, dependiendo del proveedor puedes verlo con el comando, podría ser la interfaz eth0 en algunos casos:

ifconfig -a

```
root@valentina-VirtualBox: /home/valentina
File Edit View Search Terminal Help
(Reading database ... 115800 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20161116.90da8a0-1ubuntu1_amd64.deb .
..
Unpacking net-tools (1.60+git20161116.90da8a0-1ubuntu1) ...
Setting up net-tools (1.60+git20161116.90da8a0-1ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1)
root@valentina-VirtualBox: /home/valentina# ifconfig -a
docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    inet6 fe80::42:cbff:fe0e:765 prefixlen 64 scopeid 0x20<link>
    ether 02:42:cb:0e:07:65 txqueuelen 0 (Ethernet)
    RX packets 4337 bytes 187214 (187.2 KB)
```

y buscar el IP público o bien con el IP público asignado por tu proveedor de la nube, solo debes acceder desde el navegador como

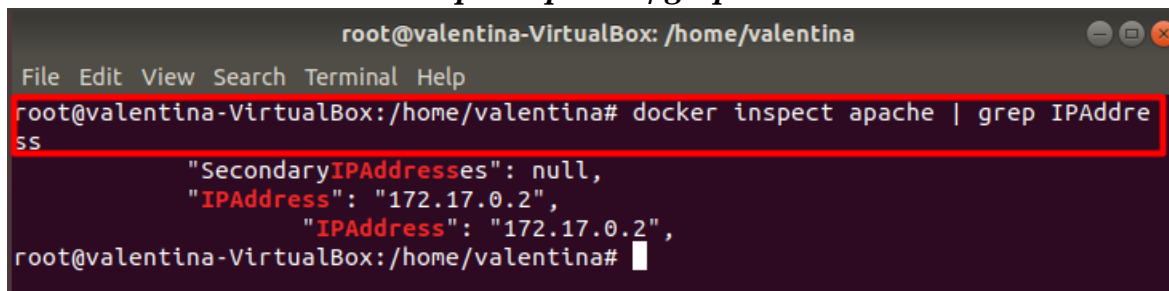
http://IP:PORT



Paso 4: Ver IP del container

En algunos casos, como prueba local podrás acceder al IP del container, ya sea desde adentro del mismo con ifconfig o bien con el comando docker en otra consola ejecutando:

docker inspect apache | grep IPAdress

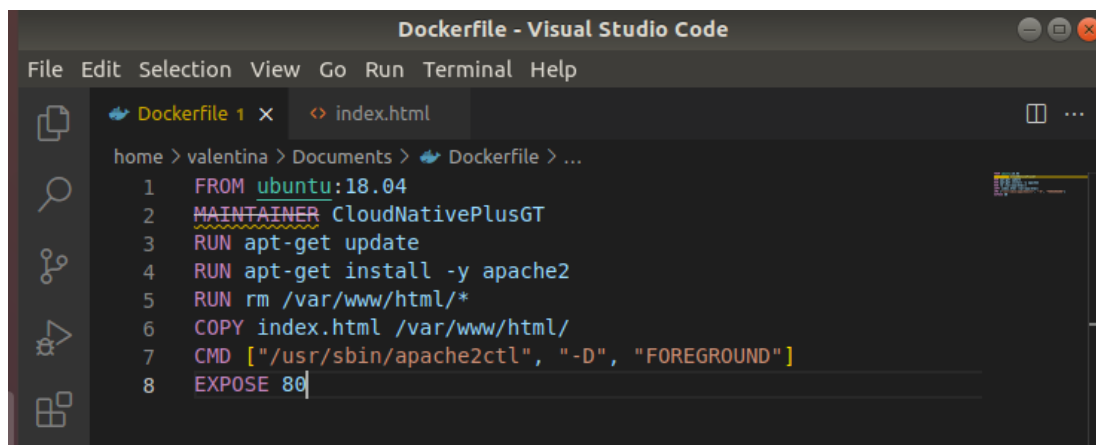


```
root@valentina-VirtualBox: /home/valentina
File Edit View Search Terminal Help
root@valentina-VirtualBox:/home/valentina# docker inspect apache | grep IPAdress
"SecondaryIPAddresses": null,
  "IPAddress": "172.17.0.2",
    "IPAddress": "172.17.0.2",
root@valentina-VirtualBox:/home/valentina#
```

Taller 3: Creación de imágenes de containers con Dockerfiles

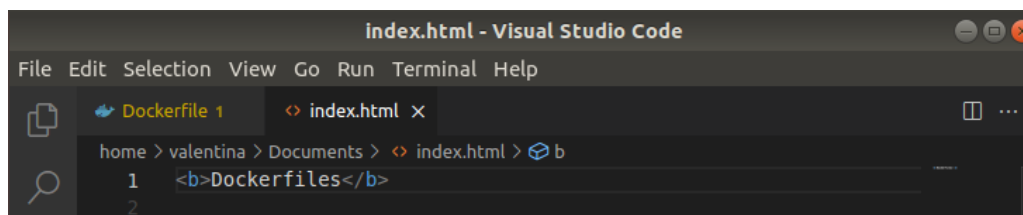
Paso 1: Crear el archivo Dockerfile e index.html

Primero debes crear el archivo llamado "Dockerfile" con el siguiente contenido:



```
Dockerfile - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Dockerfile 1 x index.html
home > valentina > Documents > Dockerfile > ...
1 FROM ubuntu:18.04
2 MAINTAINER CloudNativePlusGT
3 RUN apt-get update
4 RUN apt-get install -y apache2
5 RUN rm /var/www/html/*
6 COPY index.html /var/www/html/
7 CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
8 EXPOSE 80
```

Seguidamente debes crear el archivo "index.html" con una línea de texto

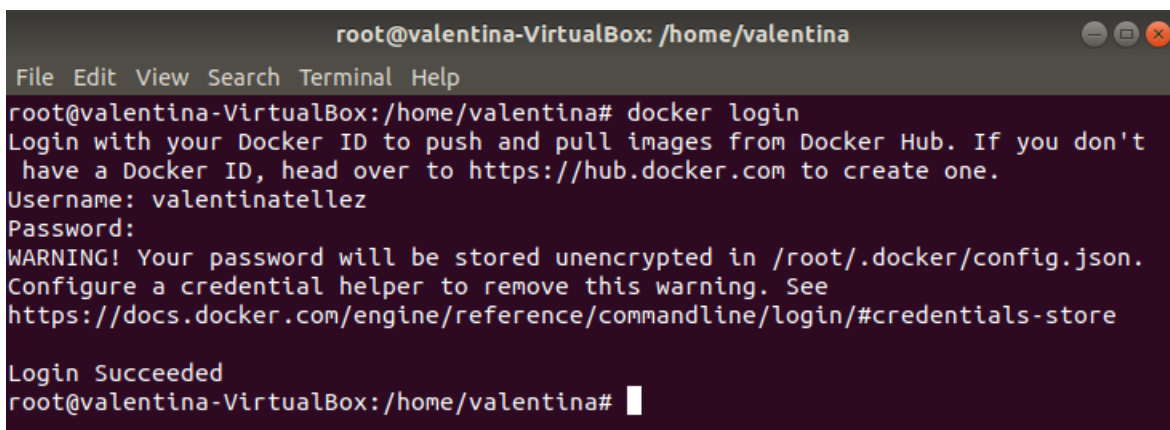


```
index.html - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Dockerfile 1 index.html x
home > valentina > Documents > index.html > b
1 <b>Dockerfiles</b>
2
```

Paso 2: Construir imagen y subirla a DockerHub para que sea pública

Primero nos logueamos con docker login, ingresando nuestro usuario y contraseña creados en el DockerHub:

docker login

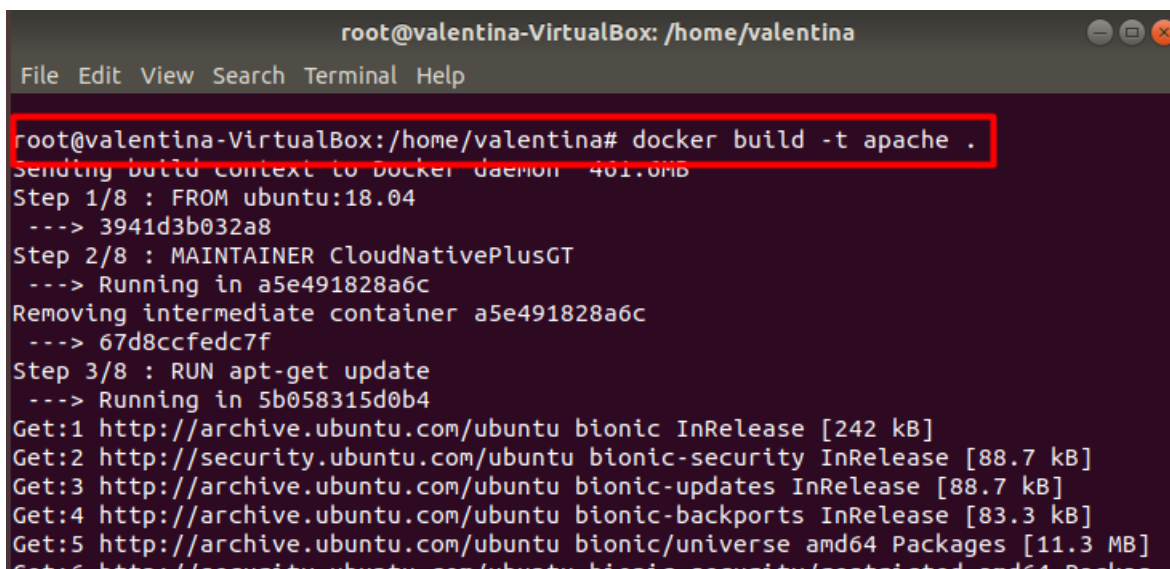


```
root@valentina-VirtualBox: /home/valentina
File Edit View Search Terminal Help
root@valentina-VirtualBox:/home/valentina# docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't
have a Docker ID, head over to https://hub.docker.com to create one.
Username: valentinatellez
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@valentina-VirtualBox:/home/valentina#
```

Luego ejecutamos como root el comando docker build para construir la imagen, en esta caso si utilizamos docker images nos mostrará que se acaba de crear una imagen llamada apache, si deseamos borrarla y no esta en uso podemos utilizar el comando docker rmi, si esta en uso debemos borrar los containers asociados a esa imagen

docker build -t apache .



```
root@valentina-VirtualBox: /home/valentina
File Edit View Search Terminal Help
root@valentina-VirtualBox:/home/valentina# docker build -t apache .
Sending build context to Docker daemon 401.0MB
Step 1/8 : FROM ubuntu:18.04
--> 3941d3b032a8
Step 2/8 : MAINTAINER CloudNativePlusGT
--> Running in a5e491828a6c
Removing intermediate container a5e491828a6c
--> 67d8ccfedc7f
Step 3/8 : RUN apt-get update
--> Running in 5b058315d0b4
Get:1 http://archive.ubuntu.com/ubuntu bionic InRelease [242 kB]
Get:2 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [83.3 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [11.3 MB]
Get:6 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages
```

Luego tagueamos la imagen local apache con nuestro usuario y el repositorio llamado apache(usuarioDockerHub/apache)

docker tag apache usuarioDockerHub/apache

```
Successfully built a7d8ee64f1de
Successfully tagged apache:latest
root@valentina-VirtualBox:/home/valentina# docker tag apache valentinatellez/ap
ache
root@valentina-VirtualBox:/home/valentina#
```

Ahora subimos la imagen al repositorio

docker push usuarioDockerHub/apache

```
> Running in b19f7503e199
Removing intermediate container b19f7503e199
--> 6fe293a7c1c6
Step 8/8 : EXPOSE 80
--> Running in 21a0f5d83157
Removing intermediate container 21a0f5d83157
--> a7d8ee64f1de
Successfully built a7d8ee64f1de
Successfully tagged apache:latest
root@valentina-VirtualBox:/home/valentina# docker tag apache valentinatellez/ap
ache
root@valentina-VirtualBox:/home/valentina# docker push valentinatellez/apache
Using default tag: latest
The push refers to repository [docker.io/valentinatellez/apache]
c767fdc1ff20: Pushed
06dc3d3776c7: Pushed
7fda09da14eb: Pushed
febe25eead6d: Pushed
b7e0fa7bfe7f: Mounted from library/ubuntu
latest: digest: sha256:c4ab579be18cdaac07f0cfe5bd7392398ea91315456f9fad10a46a34
fb85ad83 size: 1367
```

Si deseas puedes usar el script build.sh dandole permisos de ejecución al mismo, ejecutandoló y mandando de parámetro el nombre del usuario del DockerHub(Debes estar en la carpeta del Dockerfile)



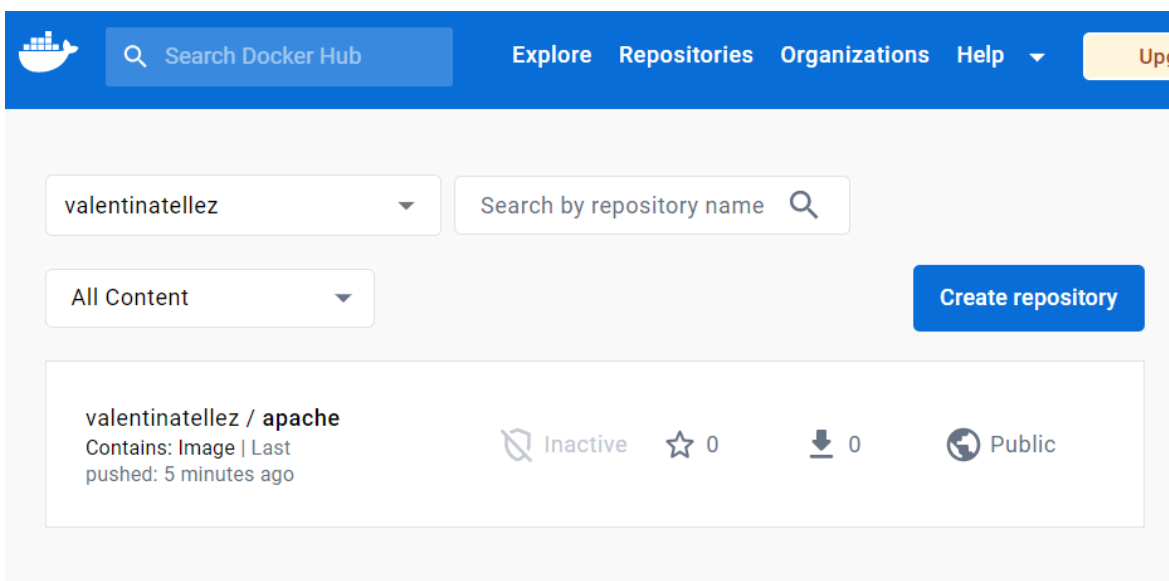
```
Open build.sh Save
chmod 777 build.sh
./build valentinatellez
```

Paso 3: Utiliza la imagen creada

En dado caso la imagen sea privada debes de logearte con el usuario correspondiente para poder descargarla. Para poder usarla ejecutarás el siguiente comando:

```
docker run -it -d -p 8888:80 --name=apacheNuevo  
usuarioDockerHub/apache
```

```
root@valentina-VirtualBox: /home/valentina
File Edit View Search Terminal Help
root@valentina-VirtualBox:/home/valentina# docker run -it -d -p 8888:80 --name=
apacheNuevo valentinatellez/apache
896983989a4efbe403c2f6b68746951376873e52789a90e77c5896dfb27640aa
root@valentina-VirtualBox:/home/valentina#
```



Paso 4: Monitoreo

Para monitorear tu container puedes ejecutar el siguiente comando:

```
docker stats apacheNuevo
```

```
root@valentina-VirtualBox: /home/valentina
File Edit View Search Terminal Help
CONTAINER ID   NAME          CPU %       MEM USAGE / LIMIT   MEM %      NET I/O
896983989a4e   apacheNuevo   0.00%       6.418MiB / 1.941GiB  0.32%      3.74kB /
0B            3.58MB / 0B   56
```

Taller 4: Uso de Docker Compose

Paso 1: Descargar Docker Compose e instalarlo

apt install Docker-compose

```

root@valentina-VirtualBox: /home/valentina
File Edit View Search Terminal Help
root@valentina-VirtualBox:/home/valentina# apt install docker-compose
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
 fonts-liberation2 fonts-opensymbol gir1.2-gst-plugins-base-1.0
 gir1.2-gstreamer-1.0 gir1.2-gudev-1.0 gir1.2-udisks-2.0
 grilo-plugins-0.3-base gstreamer1.0-gtk3 libboost-date-time1.65.1
 libboost-filesystem1.65.1 libboost-iostreams1.65.1 libboost-locale1.65.1
 libcdr-0.1-1 libclucene-contribs1v5 libclucene-core1v5 libcmis-0.5-5v5
 libcolamd2 libdazzle-1.0-0 libe-book-0.1-1 libdataserverui-1.2-2 libeot

```

Ahora para probar que podemos accederlo ejecutamos:

`docker-compose --version`

```
root@valentina-VirtualBox:/home/valentina# docker-compose --version
docker-compose version 1.17.1, build unknown
root@valentina-VirtualBox:/home/valentina#
```

Paso 2: Crear el archivo docker-compose.yml para agrupar microservicios

Puedes crear un archivo llamado `docker-compose.yml` en tu ruta actual con el siguiente contenido:

```
Open  docker-compose.yml  Save  [Menu]  [Close]

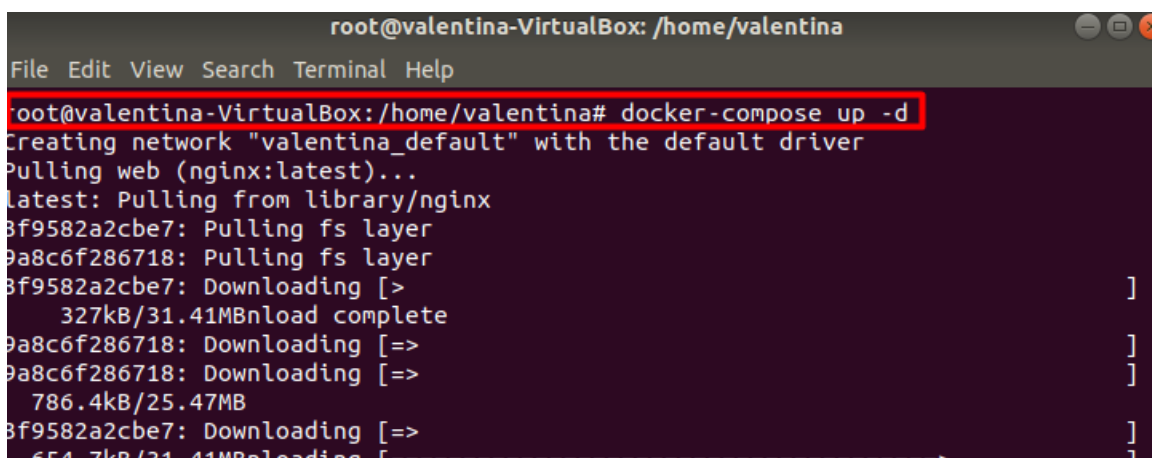
version: '3'
services:
  web:
    image: "nginx"
    ports:
      - "8000:80"
  redis:
    image: "redis"
```

Este archivo crea 2 containers, uno con el servidor web nginx y otro con una base de datos nosql clave valor redis, nginx se expone en el puerto local 8000 redireccionandolo al puerto 80 del container.

Paso 3: Administrar los microservicios con Docker Compose

Dentro de la carpeta que contiene le archivo docker-compose.yml ejecutamos:

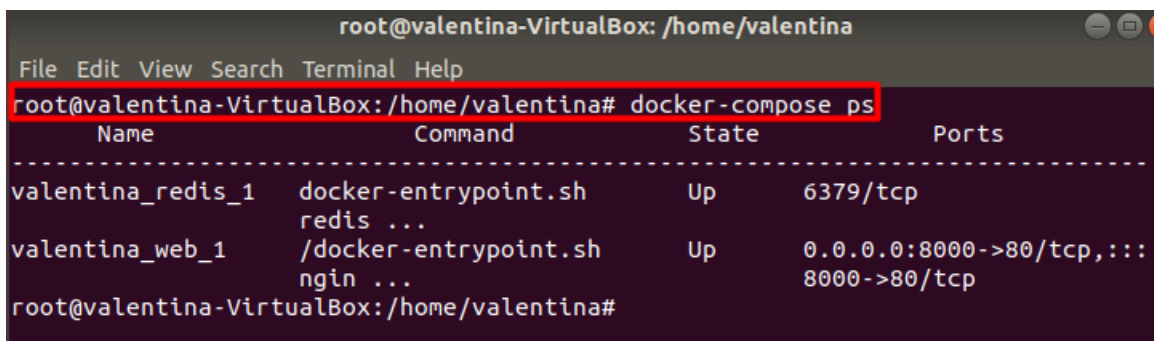
docker-compose up -d



```
root@valentina-VirtualBox: /home/valentina
File Edit View Search Terminal Help
root@valentina-VirtualBox:/home/valentina# docker-compose up -d
Creating network "valentina_default" with the default driver
Pulling web (nginx:latest)...
latest: Pulling from library/nginx
3f9582a2cbe7: Pulling fs layer
9a8c6f286718: Pulling fs layer
3f9582a2cbe7: Downloading [>]
327kB/31.41MBnload complete
9a8c6f286718: Downloading [=>]
9a8c6f286718: Downloading [=>]
786.4kB/25.47MB
3f9582a2cbe7: Downloading [=>]
654.7kB/31.41MBloading [=====>]
```

Este comando creara estos 2 containers en modo detach es decir en segundo plano, para poder ejecutar otros comandos para verificar si fueron creados exitosamente, para esto ejecuta:

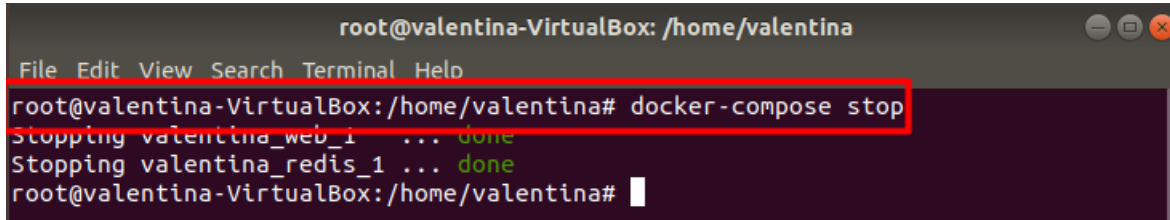
docker-compose ps



```
root@valentina-VirtualBox: /home/valentina
File Edit View Search Terminal Help
root@valentina-VirtualBox:/home/valentina# docker-compose ps
      Name                    Command                    State      Ports
-----
valentina_redis_1  docker-entrypoint.sh      Up        6379/tcp
                  redis ...
valentina_web_1    /docker-entrypoint.sh      Up        0.0.0.0:8000->80/tcp, :::
                  nginx ...                  8000->80/tcp
root@valentina-VirtualBox:/home/valentina#
```

Ahora bien si deseas detener estos containers, debes de ejecutar el siguiente comando:

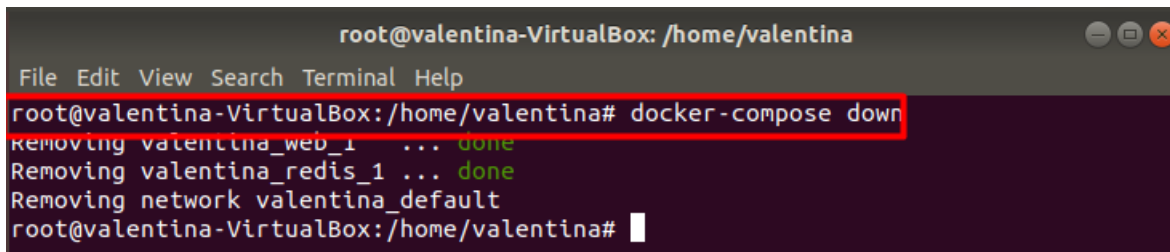
docker-compose stop

A terminal window titled 'root@valentina-VirtualBox: /home/valentina' with a menu bar (File, Edit, View, Search, Terminal, Help). The command 'docker-compose stop' is entered and highlighted with a red box. The output shows 'Stopping valentina_web_1 ... done' and 'Stopping valentina_redis_1 ... done' in green text. The prompt returns to 'root@valentina-VirtualBox: /home/valentina#'.

```
root@valentina-VirtualBox: /home/valentina
File Edit View Search Terminal Help
root@valentina-VirtualBox:/home/valentina# docker-compose stop
Stopping valentina_web_1 ... done
Stopping valentina_redis_1 ... done
root@valentina-VirtualBox:/home/valentina#
```

Si deseas detener y borrar todo ejecuta:

docker-compose down

A terminal window titled 'root@valentina-VirtualBox: /home/valentina' with a menu bar (File, Edit, View, Search, Terminal, Help). The command 'docker-compose down' is entered and highlighted with a red box. The output shows 'Removing valentina_web_1 ... done', 'Removing valentina_redis_1 ... done', and 'Removing network valentina_default' in green text. The prompt returns to 'root@valentina-VirtualBox: /home/valentina#'.

```
root@valentina-VirtualBox: /home/valentina
File Edit View Search Terminal Help
root@valentina-VirtualBox:/home/valentina# docker-compose down
Removing valentina_web_1 ... done
Removing valentina_redis_1 ... done
Removing network valentina_default
root@valentina-VirtualBox:/home/valentina#
```