

Gestión de Contenedores Dockers

LAB Gestión de Contenedores con Docker en Windows 10 Pro

Objetivos

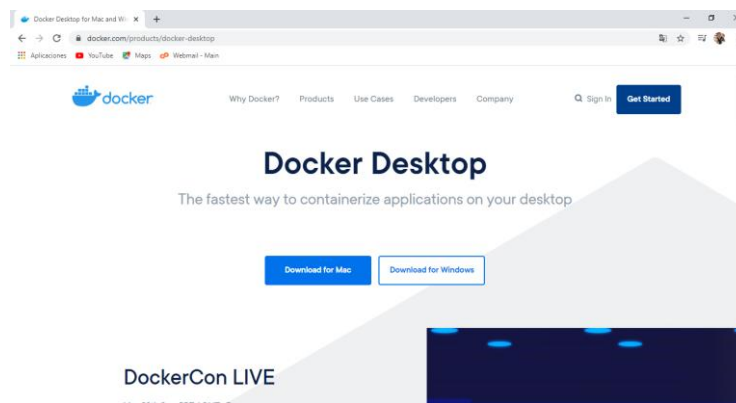
- Mostrar al participante el procedimiento para la instalación del software de gestión de contenedores Docker en Windows

Requisito

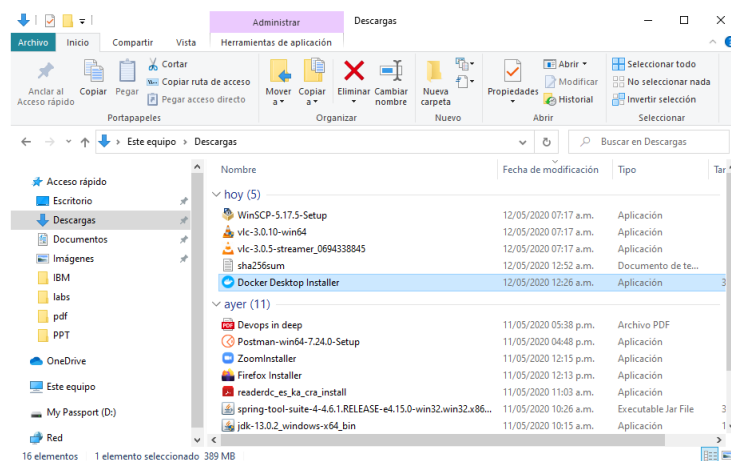
- Tener instalado el sistema operativo Windows 10 Pro/Home

Procedimiento

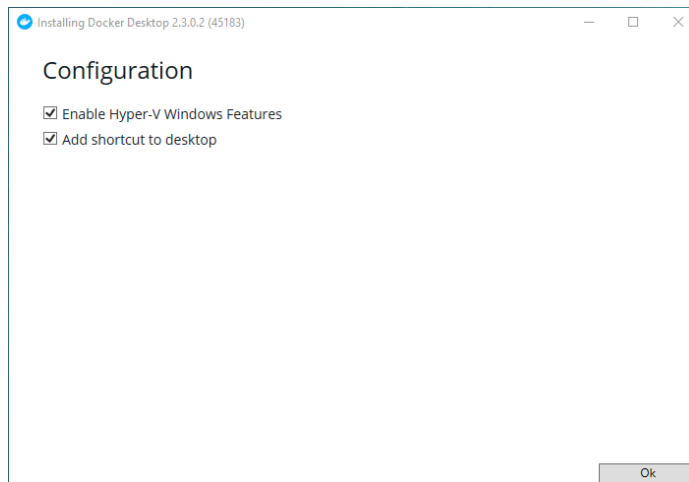
1. Descarga el software Docker para Windows desde <https://www.docker.com/products/docker-desktop>



2. Ejecuta el instalador “**Docker Desktop Installer**”, como administrador.



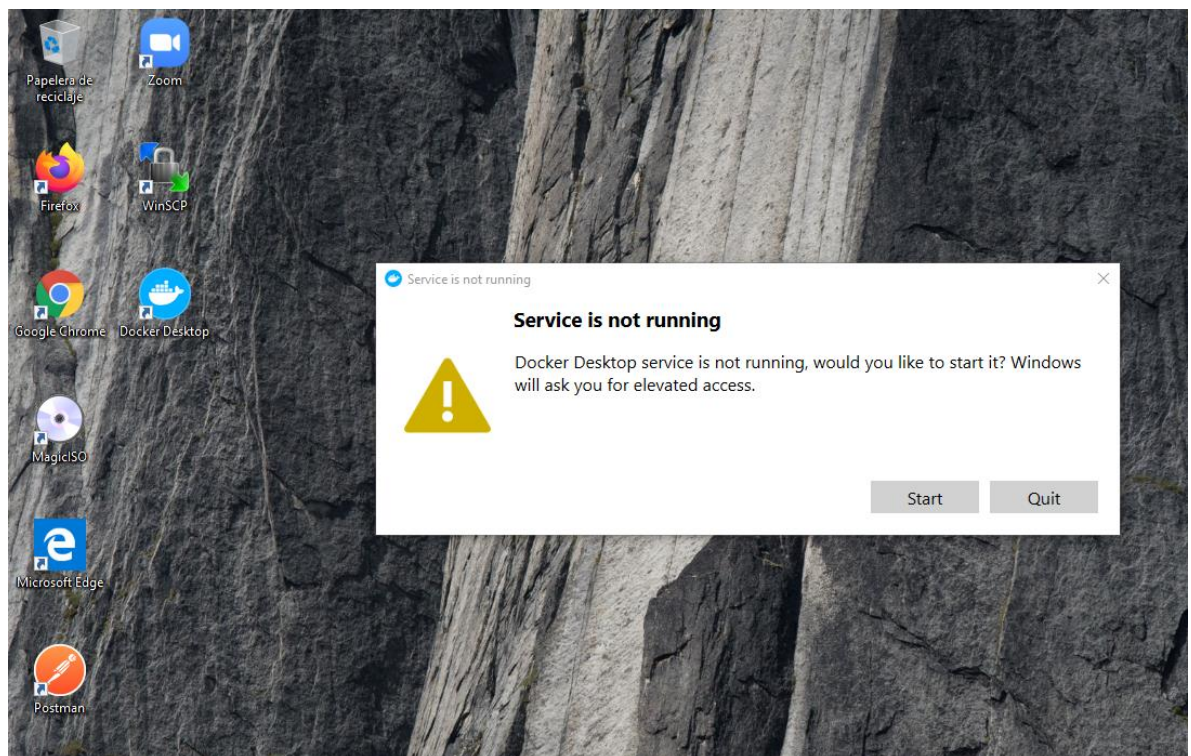
Sigues los pasos del asistente, acepta las opciones por defecto.



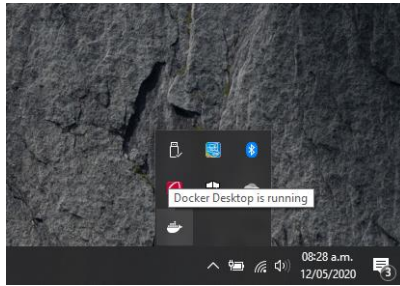
Nota: A diferencia de versiones anteriores al 2020, Docker en Windows ahora tiene soporte de virtualización nativa en Windows con **Hyper-V**.

Espera a que termine la instalación y luego reinicia la PC.

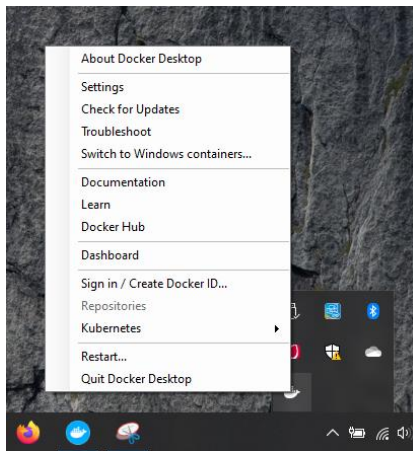
3. Iniciando el servicio Docker. Haz clic en icono del escritorio “**Docker Desktop**” e inicia el servicio de Docker.



Haz clic en **Start**. Verifica el estado del pasando el puntero del mouse por sobre el icono de Docker.



Puedes acceder a la configuración, haciendo clic en el icono Docker en la barra de Windows y seleccionando “**Settings**”.



Usando una consola **PowerShell** también podemos conocer el estado de Docker. En la consola ejecuta: `C:\> docker info`

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\daddy> docker info
Client:
 Debug Mode: false

Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 0
 Server Version: 19.03.8
 Storage Driver: overlay2
  Backing Filesystem: <unknown>
  Supports d_type: true
  Native Overlay Diff: true
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
 Swarm: inactive
 Runtimes: runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: 7ad184331fa3e55e52b890ea09e65ba581ae3429
 runc version: dc9208a3303feef5b3839f4323d9beb36df0a9dd
 init version: fec3683
 Security Options:
  seccomp
   Profile: default
 Kernel Version: 4.19.76-linuxkit
 Operating System: Docker Desktop
 OSType: linux
 Architecture: x86_64
 CPUs: 2
 Total Memory: 1.945GiB
```

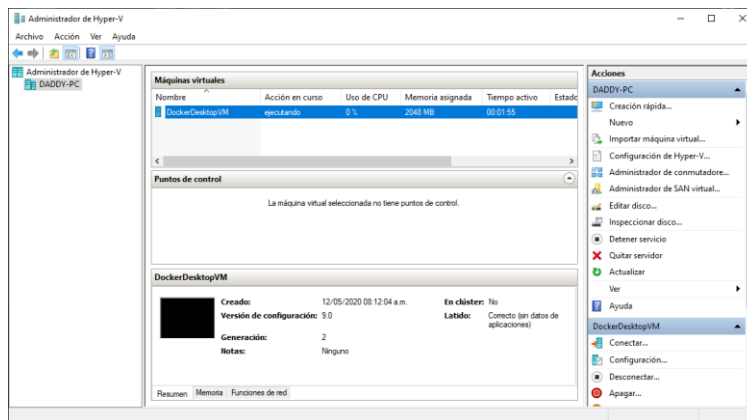
Conociendo la versión de Docker, ejecuta `c:\> docker version`

```
Windows PowerShell
Debug Mode: true
File Descriptors: 39
Goroutines: 50
System Time: 2020-05-12T13:34:13.712796176Z
EventListeners: 3
Registry: https://index.docker.io/v1/
Labels:
Experimental: false
Insecure Registries:
  127.0.0.0/8
Live Restore Enabled: false
Product License: Community Engine

PS C:\Users\daddy> docker version
Client: Docker Engine - Community
Version: 19.03.8
API version: 1.40
Go version: go1.12.17
Git commit: afacb8b
Built: Wed Mar 11 01:23:10 2020
OS/Arch: windows/amd64
Experimental: false

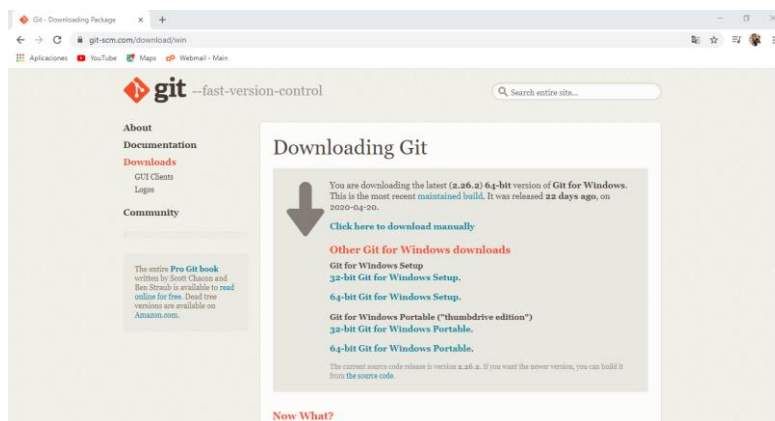
Server: Docker Engine - Community
Engine:
  Version: 19.03.8
  API version: 1.40 (minimum version 1.12)
  Go version: go1.12.17
  Git commit: afacb8b
  Built: Wed Mar 11 01:29:16 2020
  OS/Arch: linux/amd64
  Experimental: false
containerd:
  Version: v1.2.13
  GitCommit: 7ad184331fa3e55e52b890ea95e65ba581ae3429
runc:
  Version: 1.0.0-rc10
  GitCommit: dc9208a3303feef5b3839f4323d9beb36df0a9dd
docker-init:
  Version: 0.18.0
  GitCommit: fec3683
PS C:\Users\daddy>
```

4. Verifica la maquina virtual creada en Hyper-V.

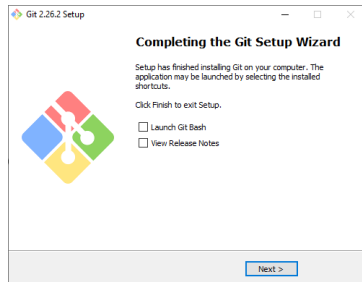


Nota: Durante la instalación se ha creado la máquina virtual “**DockerDesktopVM**” para la ejecución de Docker.

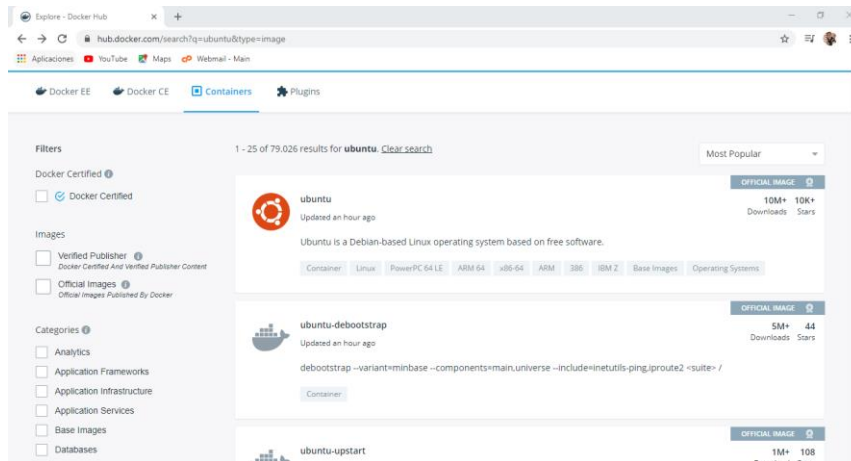
5. Instala Git en Windows. Descarga Git desde <https://git-scm.com/download/win>



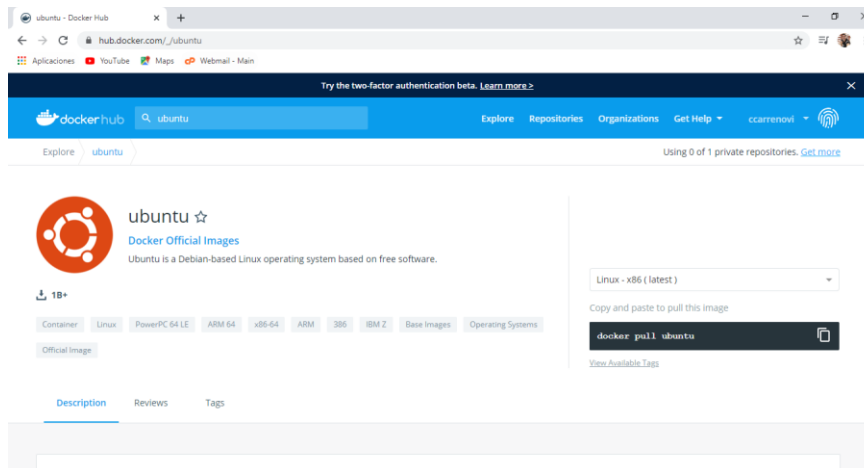
Ejecuta el instalador de Git y acepta todas las opciones por defecto.



6. Gestión de imágenes de Docker. Abre <https://hub.docker.com/> y busca la imagen de Ubuntu ubica la imagen oficial.

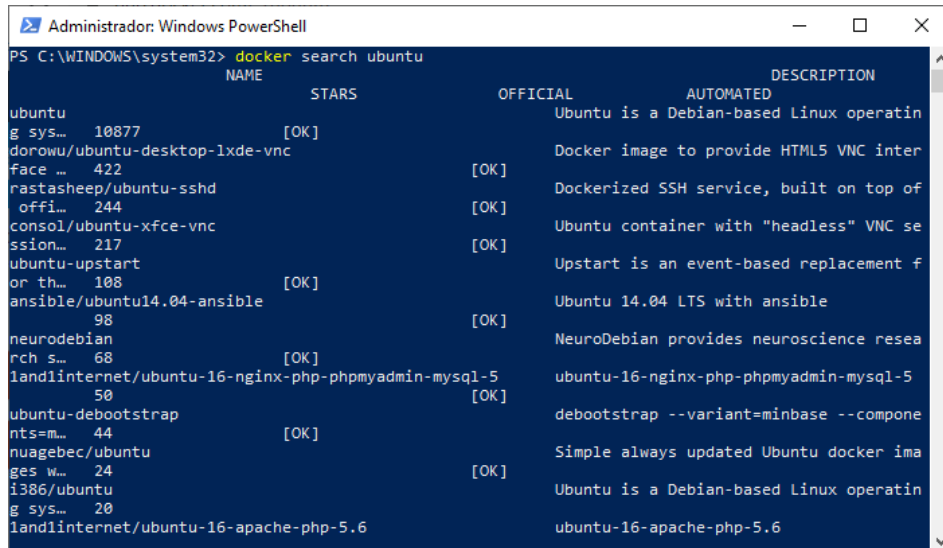


Ingresa al ítem Ubuntu “OFICIAL IMAGE”.



Anota el comando para la descarga de la imagen: **docker pull ubuntu**

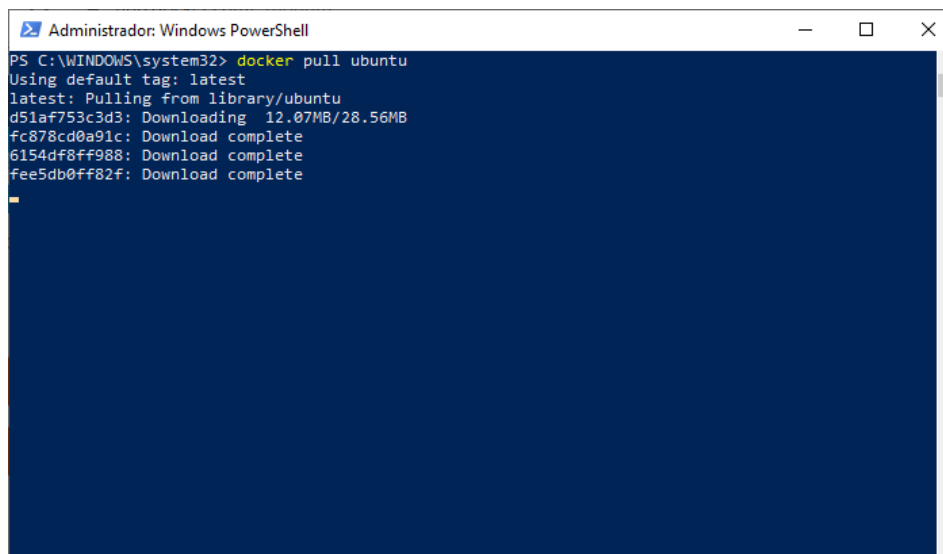
- Abre una consola PowerShell y ejecuta el comando para la descarga de la imagen de Ubuntu. Primero buscamos la imagen con el comando: `docker search ubuntu`



```
Administrador: Windows PowerShell
PS C:\WINDOWS\system32> docker search ubuntu
```

NAME	STARS	OFFICIAL	AUTOMATED	DESCRIPTION
ubuntu	10877	[OK]		Ubuntu is a Debian-based Linux operatin
g sys...	422	[OK]		Docker image to provide HTML5 VNC inter
dorowu/ubuntu-desktop-lxde-vnc	244	[OK]		Dockerized SSH service, built on top of
face ...	217	[OK]		Ubuntu container with "headless" VNC se
rastasheep/ubuntu-sshd	108	[OK]		Upstart is an event-based replacement f
offi...	98	[OK]		Ubuntu 14.04 LTS with ansible
consol/ubuntu-xfce-vnc	68	[OK]		NeuroDebian provides neuroscience resea
ssion...	50	[OK]		ubuntu-16-nginx-php-phpmyadmin-mysql-5
ubuntu-upstart	44	[OK]		debootstrap --variant=minbase --compone
or th...	24	[OK]		Simple always updated Ubuntu docker ima
ansible/ubuntu14.04-ansible	20	[OK]		Ubuntu is a Debian-based Linux operatin
neurodebian				ubuntu-16-apache-php-5.6
rch s...				
landinternet/ubuntu-16-nginx-php-phpmyadmin-mysql-5				
ubuntu-debootstrap				
nts=m...				
nuagebec/ubuntu				
ges w...				
i386/ubuntu				
g sys...				
landinternet/ubuntu-16-apache-php-5.6				

Luego ejecuta: `docker pull ubuntu`



```
Administrador: Windows PowerShell
PS C:\WINDOWS\system32> docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
d51af753c3d3: Downloading 12.07MB/28.56MB
fc878cd0a91c: Download complete
6154df8ff988: Download complete
fee5db0ff82f: Download complete
```

Espera a que descargue la imagen.

- Verifica las imágenes descargadas. Para listar las imágenes en el sistema ejecuta: `docker images`

Nota: Observa que cada imagen tiene un identificador llamado “**IMAGE ID**”, el cual contiene un valor hexadecimal único. También el campo “**tag**” es importante en este caso es la ultima version de la imagen “latest”.

```
Administrador: Windows PowerShell
PS C:\WINDOWS\system32> docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
d51af753c3d3: Pull complete
fc878cd0a91c: Pull complete
6154df8ff988: Pull complete
fee5db0ff82f: Pull complete
Digest: sha256:747d2dbbaee995098c9792d99bd333c6783ce56150d1b11e333bbceed5c54d7
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
PS C:\WINDOWS\system32> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              latest             1d622ef86b13       2 weeks ago        73.9MB
PS C:\WINDOWS\system32>
```

9. Usando tags para descargar una version especifica de la imagen. Descarguemos la imagen de ubuntu en version 18.10

```
Administrador: Windows PowerShell
PS C:\WINDOWS\system32> docker search ubuntu:18.10
NAME                DESCRIPTION                STARS     OFFICIAL   AUTOMATED
amwoolterton/zookeeper-kubernetes  zookeeper-3.4.13 / ubuntu:18.10    1
topt/python3        python3 environment based on ubuntu:18.10    0
joegagliardo/ubuntu This is an Ubuntu image based on ubuntu:18.1... 0
kutim/ubuntu        clone ubuntu:18.10          0
PS C:\WINDOWS\system32> docker pull ubuntu:18.10
18.10: Pulling from library/ubuntu
8a532469799e: Pull complete
f4dcec3531: Pull complete
0701585e: Pull complete
622967: Pull complete
sha256:7d657275047118bb77b052c4c0ae43e8a289ca2879ebfa78a703c93aa8fd686c
Status: Downloaded newer image for ubuntu:18.10
docker.io/library/ubuntu:18.10
PS C:\WINDOWS\system32> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              latest             1d622ef86b13       2 weeks ago        73.9MB
ubuntu              18.10             9dc19675e327       9 months ago       67.3MB
PS C:\WINDOWS\system32>
```

10. Creando nuestro primer contenedor “for Dummy”, será nuestro contenedor hello-world. Crearemos un contenedor que emite un mensaje simple “**Hello from Docker!**”.

Nota: Es necesario recordar que cuando Docker ejecuta un contenedor y no existe la imagen, el controlador Docker intentara descargar la imagen apropiada.

Para iniciar un contenedor ejecuta el comando C:\> **docker run** <contenedor-name> .

Si deseas listar los contenedores que se están corriendo en el sistema, debes ejecutar el comando: C:\> **docker ps**

Algunos contenedores ejecutan una instrucción especifica y terminan el proceso, para listar los procesos de los contenedores incluyendo los que han finalizado ejecuta el comando: C:\> **docker ps -a**

```
Administrador: Windows PowerShell
PS C:\WINDOWS\system32> docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
0e03bdcc26d7: Pull complete
Digest: sha256:8e3114318a995a1ee497790535e7b88365222a21771ae7e53687ad76563e8e76
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

PS C:\WINDOWS\system32>
```

Conociendo el estado de los procesos de los contenedores.

```
Administrador: Windows PowerShell
PS C:\WINDOWS\system32> docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
PS C:\WINDOWS\system32> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
6b58b4337be5   hello-world  "/hello"                7 minutes ago   Exited (0) 7 minutes ago
```

11. Ejecutar comando en contenedores. Para ejecutar comandos en el contenedor debes iniciar un Shell o también puedes ejecutar un comando específico. Crea un contenedor usando la imagen de ubuntu.

```
Administrador: Windows PowerShell
PS C:\WINDOWS\system32> docker run ubuntu ls
bin
boot
dev
etc
home
lib
lib32
lib64
libx32
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
PS C:\WINDOWS\system32>
```

Este contenedor ejecuta el comando ls (muestra el contenido de los archivos y directorios) y termina.


```
Administrador: Windows PowerShell
PS C:\WINDOWS\system32> docker run -i -t ubuntu bash
root@c6ac412201d2:/# pwd
/
root@c6ac412201d2:/# uname -a
Linux c6ac412201d2 4.19.76-linuxkit #1 SMP Fri Apr 3 15:53:26 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
root@c6ac412201d2:/# date
Tue May 12 16:29:07 UTC 2020
root@c6ac412201d2:/# ls
bin  dev  home  lib32  libx32  mnt  proc  run  srv  tmp  var
boot  etc  lib  lib64  media  opt  root  sbin  sys  usr
root@c6ac412201d2:/# exit
exit
PS C:\WINDOWS\system32>
```

En este caso creamos un contenedor abriendo un terminal y ejecutando el Shell **bash** de **ubuntu**.

12. Eliminando un contenedor. Para eliminar un contenedor ejecuta el comando `c:\> docker rm <id contenedor>`

```
Administrador: Windows PowerShell
PS C:\WINDOWS\system32> docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
c6ac412201d2        ubuntu             "bash"             2 minutes ago      Exited (0)          About a minute ago
e1ec661fe86d        ubuntu             "determined_burnell" 5 minutes ago      Exited (0)          5 minutes ago ago
6b58b4337be5        hello-world        "/hello"           29 minutes ago     Exited (0)          29 minutes ago ago
PS C:\WINDOWS\system32> docker rm 6b
6b
PS C:\WINDOWS\system32> docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
c6ac412201d2        ubuntu             "bash"             5 minutes ago      Exited (0)          5 minutes ago ago
e1ec661fe86d        ubuntu             "determined_burnell" 9 minutes ago      Exited (0)          9 minutes ago ago
loving_dirac
```

13. Mostrando la ayuda de los comandos en Docker. Para mostrar la ayuda de los comandos ejecuta el comando con la opción **--help**

```
Administrador: Windows PowerShell
PS C:\WINDOWS\system32> docker images --help
Usage: docker images [OPTIONS] [REPOSITORY[:TAG]]

List images

Options:
  -a, --all            Show all images (default hides intermediate images)
  --digests            Show digests
  -f, --filter filter  Filter output based on conditions provided
  --format string      Pretty-print images using a Go template
  --no-trunc           Don't truncate output
  -q, --quiet          Only show numeric IDs
PS C:\WINDOWS\system32> docker images -q
1d622ef86b13
bf756fb1ae65
9dc19675e327
PS C:\WINDOWS\system32>
```

14. Salir del contenedor sin terminar la ejecución. Inicia el contenedor con la opción `-it` (interactive y un pseudo terminal) y ejecuta la combinación de teclas `[Ctrl] + [p] + [q]`

```
Administrador: Windows PowerShell
PS C:\WINDOWS\system32> docker run -it ubuntu
root@6b7f9ecdba98:/#
PS C:\WINDOWS\system32> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
6b7f9ecdba98   ubuntu    "/bin/bash"             About a minute ago    Up About a minute
c6ac412201d2   ubuntu    "bash"                  38 minutes ago       Exited (0) 38 minutes ago
elec661fe86d   ubuntu    "ls"                    42 minutes ago       Exited (0) 42 minutes ago
loving_dirac
PS C:\WINDOWS\system32> docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
6b7f9ecdba98   ubuntu    "/bin/bash"             About a minute ago    Up About a minute
stoic_swartz
PS C:\WINDOWS\system32> docker stop 6b
6b
PS C:\WINDOWS\system32> docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
6b7f9ecdba98   ubuntu    "/bin/bash"             About a minute ago    Up About a minute
stoic_swartz
PS C:\WINDOWS\system32>
```

15. Contenedores interactivos. Si el contenedor fue creado con la opción `-it`, podemos interactuar con el contenedor, podemos iniciarlo con la opción `start`, podemos recuperar el “focus” con la opción `attach`.

```
Administrador: Windows PowerShell
PS C:\WINDOWS\system32> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
6b7f9ecdba98   ubuntu    "/bin/bash"             27 minutes ago    Exited (0) 11 seconds ago
c6ac412201d2   ubuntu    "bash"                  About an hour ago    Exited (0) About an hour ago
elec661fe86d   ubuntu    "ls"                    About an hour ago    Exited (0) About an hour ago
loving_dirac
PS C:\WINDOWS\system32> docker start 6b7f9ecdba98
6b7f9ecdba98
PS C:\WINDOWS\system32> docker attach 6b7f9ecdba98
root@6b7f9ecdba98:/# pwd
/
root@6b7f9ecdba98:/# uname -a
Linux 6b7f9ecdba98 4.19.76-linuxkit #1 SMP Fri Apr 3 15:53:26 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
root@6b7f9ecdba98:/# ls
bin  dev  home  lib32  libx32  root  proc  run  srv  tmp  var
boot  etc  lib  lib64  media  opt  root  sbin  sys  usr
root@6b7f9ecdba98:/# date
Tue May 12 17:34:42 UTC 2020
root@6b7f9ecdba98:/# read escape sequence
PS C:\WINDOWS\system32> docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
6b7f9ecdba98   ubuntu    "/bin/bash"             28 minutes ago    Up 58 seconds
stoic_swartz
PS C:\WINDOWS\system32>
```

16. Creando contenedores con nombre. Podemos asignar un nombre al contenedor con el parámetro `--name`

```
Administrador: Windows PowerShell
PS C:\WINDOWS\system32> docker run --name dev-ubuntu -it ubuntu
root@9e33fc24f1ad:/#
PS C:\WINDOWS\system32> docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
9e33fc24f1ad   ubuntu    "/bin/bash"             23 seconds ago    Up 21 seconds
dev-ubuntu
6b7f9ecdba98   ubuntu    "/bin/bash"             42 minutes ago    Up 14 minutes
stoic_swartz
PS C:\WINDOWS\system32>
```

17. Personalización de Contenedores. Crearemos un contenedor basado en ubuntu y realizaremos la instalación de algunas aplicaciones clásicas.

```
root@36b7729ed258: /
PS C:\WINDOWS\system32> docker run --name qa-ubuntu -it ubuntu bash
root@36b7729ed258:/# apt update
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [60.9 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [4673 B]
Get:5 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [8273 B]
Get:6 http://archive.ubuntu.com/ubuntu focal-updates InRelease [107 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:8 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [177 kB]
Get:9 http://archive.ubuntu.com/ubuntu focal/main amd64 Packages [1275 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal/restricted amd64 Packages [33.4 kB]
Get:11 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [11.3 MB]
69% [11 Packages 6722 kB/11.3 MB 59%]
70% [11 Packages 6883 kB/11.3 MB 61%]
Get:12 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [4673 B]
Get:13 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [32.3 kB]
Get:14 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [104 kB]
Get:15 http://archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [2903 B]
Fetched 13.6 MB in 46s (294 kB/s)
Reading package lists... 10%
```

En el contenedor ejecuta los comandos:

```
$ apt install nano
```

Luego:

```
$ apt install git
```

```
$ apt install iproute2
```

Ahora podemos ejecutar el comando: `ip addr show`.

```
root@36b7729ed258: /
root@36b7729ed258:/# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
2: tunl0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/ipip 0.0.0.0 brd 0.0.0.0
3: ip6tnl0@NONE: <NOARP> mtu 1452 qdisc noop state DOWN group default qlen 1000
    link/tunnel6 :: brd ::
30: eth0@if31: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:11:00:04 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.17.0.4/16 brd 172.17.255.255 scope global eth0
        valid_lft forever preferred_lft forever
root@36b7729ed258:/#
```

Instala las herramientas para ejecutar comando de networking.

```
apt install iputils-ping
```

```
root@36b7729ed258: /
Selecting previously unselected package iputils-ping.
(Reading database ... 8223 files and directories currently installed.)
Preparing to unpack .../iputils-ping_3%3a20190709-3_amd64.deb ...
Unpacking iputils-ping (3:20190709-3) ...
Setting up iputils-ping (3:20190709-3) ...
root@36b7729ed258:/# ping 172.17.0.1
PING 172.17.0.1 (172.17.0.1) 56(84) bytes of data.
64 bytes from 172.17.0.1: icmp_seq=1 ttl=64 time=0.074 ms
64 bytes from 172.17.0.1: icmp_seq=2 ttl=64 time=0.112 ms
^C
--- 172.17.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1043ms
rtt min/avg/max/mdev = 0.074/0.093/0.112/0.019 ms
root@36b7729ed258:/#
```

18. Creando imágenes a partir de contenedores. Crearemos un container iterativo llamado web-ubuntu basado en la distribución de ubuntu, luego instalaremos apache2.

```
root@385f46064422: /
PS C:\WINDOWS\system32> docker run -it --name web-ubuntu ubuntu bash
root@385f46064422:/# apt update
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [4673 B]
Get:4 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [62.7 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [9569 B]
Get:6 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [1079 B]
Get:7 http://archive.ubuntu.com/ubuntu focal-updates InRelease [107 kB]
Get:8 http://archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:9 http://archive.ubuntu.com/ubuntu focal/main amd64 Packages [1275 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [177 kB]
Get:11 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [11.3 MB]
Get:12 http://archive.ubuntu.com/ubuntu focal/restricted amd64 Packages [33.4 kB]
Get:13 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [1079 B]
Get:14 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [110 kB]
Get:15 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [34.2 kB]
Get:16 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [4673 B]
Get:17 http://archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [2903 B]
Fetched 13.6 MB in 49s (276 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
root@385f46064422:/# apt 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 ca-certificates file krb5-locales libapr1 libaprutil1 libaprutil1-dbd-sqlite
```

Dentro del contenedor, ejecuta el comando

apt update

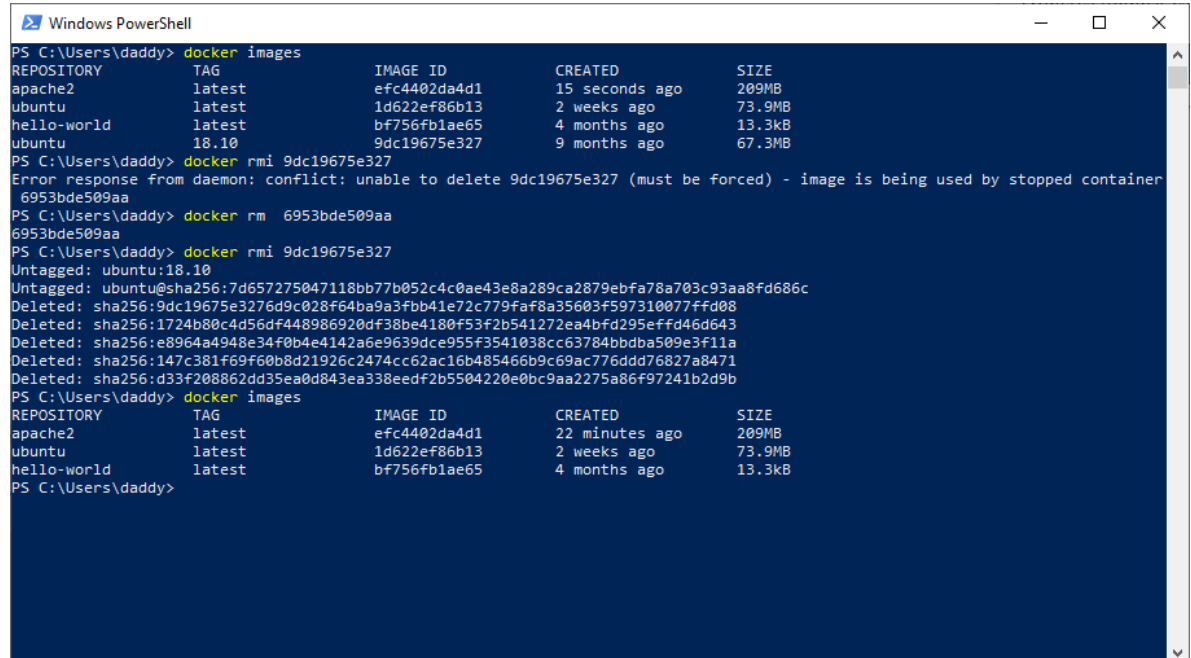
Luego:

apt install apache2

En otro terminal ubica el id del container **web-ubuntu** y crea la nueva imagen con nombre **apache2**.

```
Windows PowerShell
PS C:\Users\daddy> docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS           NAMES
385f46064422   ubuntu    "bash"                  20 minutes ago Up 20 minutes           web-ubu
9e33fc24f1ad   ubuntu    "/bin/bash"             5 hours ago   Up 5 hours           dev-ubu
6b7f9ecdba98   ubuntu    "/bin/bash"             5 hours ago   Up 5 hours           stoic_s
wartz
PS C:\Users\daddy> docker commit 385f46064422 apache2
e9c4402da4d1fc68a140ff24103547d4e97e5c7a6694a5c002f2fa8a621e7cfd
PS C:\Users\daddy> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
apache2        latest   e9c4402da4d1  15 seconds ago  209MB
ubuntu         latest   1d622ef86b13  2 weeks ago    73.9MB
hello-world    latest   bf756fblae65  4 months ago   13.3kB
ubuntu         18.10    9dc19675e327  9 months ago   67.3MB
PS C:\Users\daddy>
```

19. Eliminando imágenes. Para eliminar imágenes utiliza la acción **rmi** (remove image).



```
PS C:\Users\daddy> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
apache2              latest             efc4402da4d1       15 seconds ago     209MB
ubuntu               latest             1d622ef86b13       2 weeks ago        73.9MB
hello-world          latest             bf756fblae65       4 months ago       13.3kB
ubuntu               18.10             9dc19675e327       9 months ago       67.3MB

PS C:\Users\daddy> docker rmi 9dc19675e327
Error response from daemon: conflict: unable to delete 9dc19675e327 (must be forced) - image is being used by stopped container 6953bde509aa

PS C:\Users\daddy> docker rm 6953bde509aa
6953bde509aa

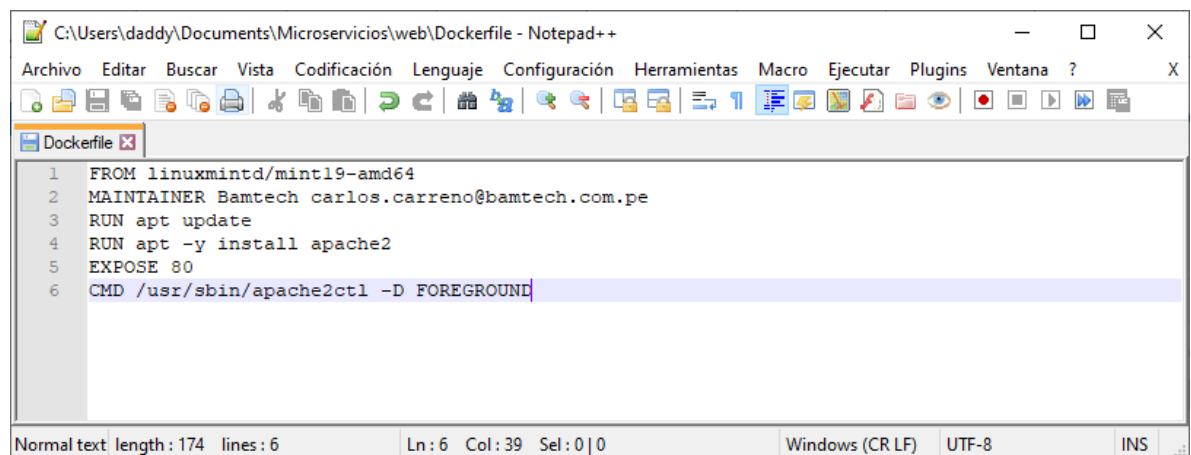
PS C:\Users\daddy> docker rmi 9dc19675e327
Untagged: ubuntu:18.10
Untagged: ubuntu@sha256:7d657275047118bb77b052c4c0ae43e8a289ca2879ebfa78a703c93aa8fd686c
Deleted: sha256:9dc19675e3276d9c028f64ba9a3fbb41e72c779faf8a35603f597310077ffd08
Deleted: sha256:1724b80c4d56df448986920df38be4180f53f2b541272ea4bfd295effd46d643
Deleted: sha256:e8964a4948e34f0b4e4142a6e9639dce955f3541038cc63784bbdba509e3f11a
Deleted: sha256:147c381f69f60b8d21926c2474cc62ac16b485466b9c69ac776ddd76827a8471
Deleted: sha256:d33f208862dd35ea0d843ea338eedf2b5504220e0bc9aa2275a86f97241b2d9b

PS C:\Users\daddy> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
apache2              latest             efc4402da4d1       22 minutes ago     209MB
ubuntu               latest             1d622ef86b13       2 weeks ago        73.9MB
hello-world          latest             bf756fblae65       4 months ago       13.3kB

PS C:\Users\daddy>
```

Nota: Observa que, si la imagen esta siendo utilizada por algún contenedor no dejara eliminarla de manera normal, primero tienes que eliminar los contenedores.

20. Creación de una imagen desde un Dockerfile. Crea el archivo de configuración de la imagen, llama al archivo **Dockerfile**.



```
C:\Users\daddy\Documents\Microservicios\web\ Dockerfile - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ventana  ?  X

1  FROM linuxmintd/mint19-amd64
2  MAINTAINER Bamtech carlos.carreno@bamtech.com.pe
3  RUN apt update
4  RUN apt -y install apache2
5  EXPOSE 80
6  CMD /usr/sbin/apache2ctl -D FOREGROUND

Normal text | length: 174 | lines: 6 | Ln: 6 | Col: 39 | Sel: 0 | 0 | Windows (CR LF) | UTF-8 | INS
```

Luego ejecuta el comando:

docker build -t web-image C:\Users\daddy\Documents\Microservicios\web\

```
Administrador: Windows PowerShell

linuxmint/mint19-amd64 latest 237cdc2aa03d 2 years ago 741MB
PS C:\WINDOWS\system32> docker build -t web-image C:\Users\daddy\Documents\Microservicios\web\
Sending build context to Docker daemon 3.072kB
Step 1/6 : FROM linuxmint/mint19-amd64
--> 237cdc2aa03d
Step 2/6 : MAINTAINER Bantech carlos.carreno@bantech.com.pe
--> Running in fbd06e52455c
Removing intermediate container fbd06e52455c
--> 6e22376238d9
Step 3/6 : RUN apt update
--> Running in 03d55f4b1b47

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Ign:2 http://packages.linuxmint.com tara InRelease
Get:3 http://packages.linuxmint.com tara Release [24.1 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic InRelease [242 kB]
Get:5 http://archive.canonical.com/ubuntu bionic InRelease [10.2 kB]
Get:6 http://packages.linuxmint.com tara Release.gpg [819 B]
Get:7 http://archive.canonical.com/ubuntu bionic/partner Sources [1,902 B]
Get:8 http://security.ubuntu.com/ubuntu bionic-security/multiverse Sources [3,233 B]
Get:9 http://security.ubuntu.com/ubuntu bionic-security/restricted Sources [5,868 B]
```

21. Creación del contenedor y exposición de puertos locales.

```
Administrador: Windows PowerShell

PS C:\WINDOWS\system32> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
web-image           latest             68116cdc66b5       31 seconds ago     805MB
ubuntu              latest            1d622ef86b13       2 weeks ago        73.9MB
hello-world         latest            bf7366fb62bd       4 months ago       13.3kB
ubuntu              18.10            9dc19675e327       9 months ago       67.3MB
linuxmint/mint19-amd64 latest            237cdc2aa03d       2 years ago        741MB
100d180744dcf87675ce421c02554aa2d5e9cef8f8fda03f4b9d4c13633
PS C:\WINDOWS\system32> docker run --name=mint-apache -d -p 8080:80 web-image
100d180744dcf87675ce421c02554aa2d5e9cef8f8fda03f4b9d4c13633
PS C:\WINDOWS\system32> docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
100d180744dc        web-image          "/bin/sh -c 'usr/sb..." 8 seconds ago       Up 6 seconds       0.0.0.0:8080->80/tcp   mint-apache
9a33fc24f1ad        ubuntu            "/bin/bash"         14 hours ago        Up 14 hours                dev-ubuntu
6b7f9ecdba98        ubuntu            "/bin/bash"         14 hours ago        Up 14 hours                stoic_swartz
```

22. Acceso al software de middleware o servicio (servicio, servidor web, servidor de aplicaciones). Para acceder al apache web server del contenedor **mint-apache**, abre un browser y acceder a la url: <http://localhost:8080>

