



INSTALACIÓN DOCKER

INSTALACIÓN DOCKER

ADRIÁN VICENTE LÓPEZ
ADRIÁN VICENTE LÓPEZ
DESPLIEGUE APlicaciones WEB
DESPLIEGUE APlicaciones WEB

2ºDAW
2ºDAW

-CONTENIDO-

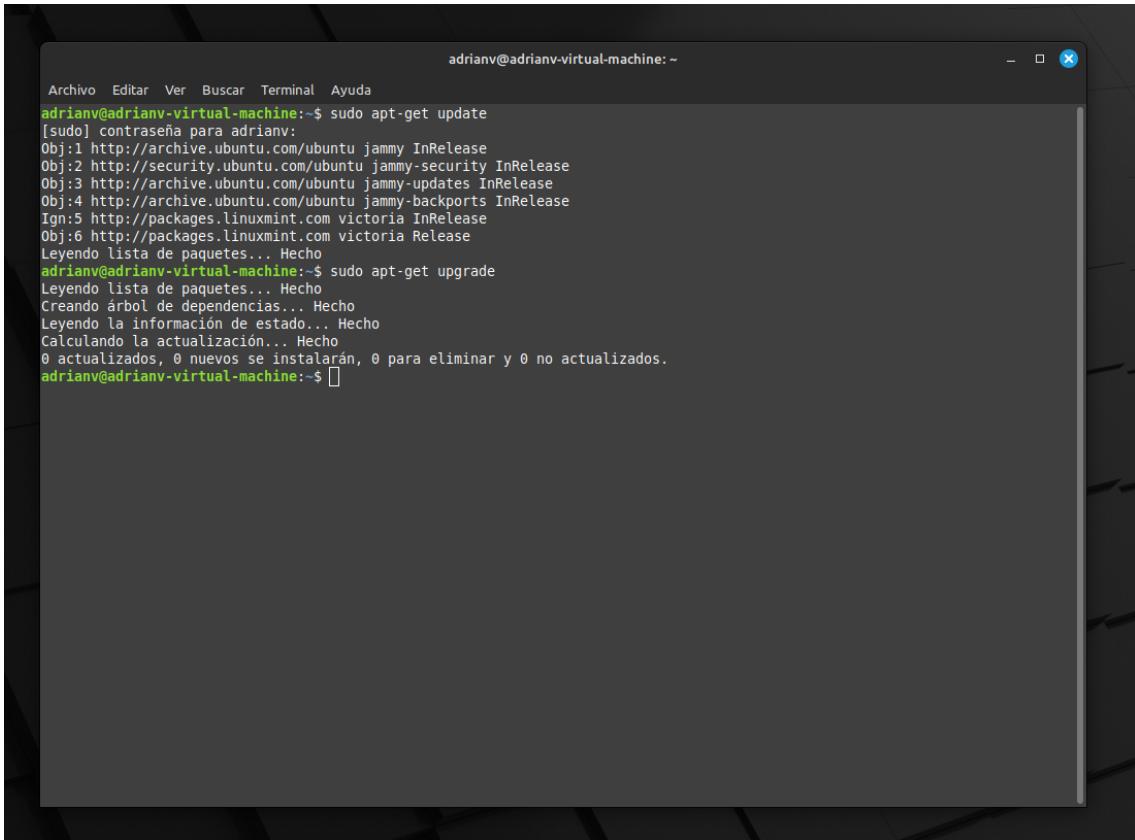
1.- INTRODUCCIÓN.....	2
2.- ACTUALIZACION DE PAQUETES EN LINUX.....	2
3.- INSTALACIÓN DE DOCKER.....	3
4.-CREANDO UN PROYECTO	4
5.-COMANDOS VARIOS.....	5
5.1 LISTAR	5
5.2 ELIMINAR IMAGEN	6
5.3 LISTAR CONTENEDORES	6
5.4 DETENER UN CONTENEDOR ESPECIFICO	6
5.5 ELIMINAR TODOS LOS CONTENEDORES DETENIDOS.....	6
5.6 MOSTRAR REGISTROS DE UN CONTENEDOR.....	6

1.- INTRODUCCIÓN

"El siguiente documento detalla el proceso de instalación de Docker, la plataforma líder para el desarrollo basado en contenedores. El objetivo de esta práctica es preparar el sistema operativo, instalar los paquetes necesarios y realizar una prueba de funcionalidad básica (Hola Mundo) para asegurar que el servicio está activo y listo para ser utilizado en el desarrollo de proyectos."

2.- ACTUALIZACION DE PAQUETES EN LINUX

aquí primeramente actualizamos los paquetes antes de la instalación de Docker.

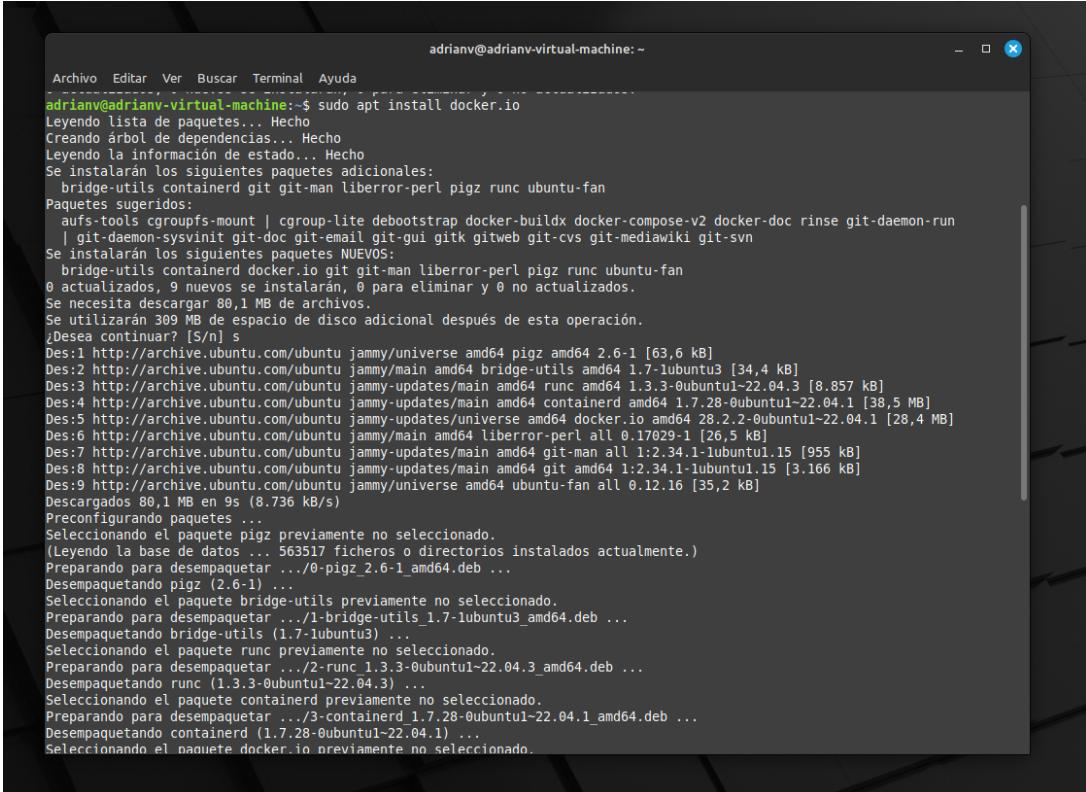


```
adrianv@adrianv-virtual-machine:~$ sudo apt-get update
[sudo] contraseña para adrianv:
Obj:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Obj:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Obj:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Obj:4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:5 http://packages.linuxmint.com victoria InRelease
Obj:6 http://packages.linuxmint.com victoria Release
Leyendo lista de paquetes... Hecho
adrianv@adrianv-virtual-machine:~$ sudo apt-get upgrade
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
adrianv@adrianv-virtual-machine:~$ 
```

Vemos en la imagen que como previamente yo ya había hecho las actualizaciones de Linux al instalar Linux pues no instala ni actualiza nada nuevo, pero, aun así, hago el comando por si acaso.

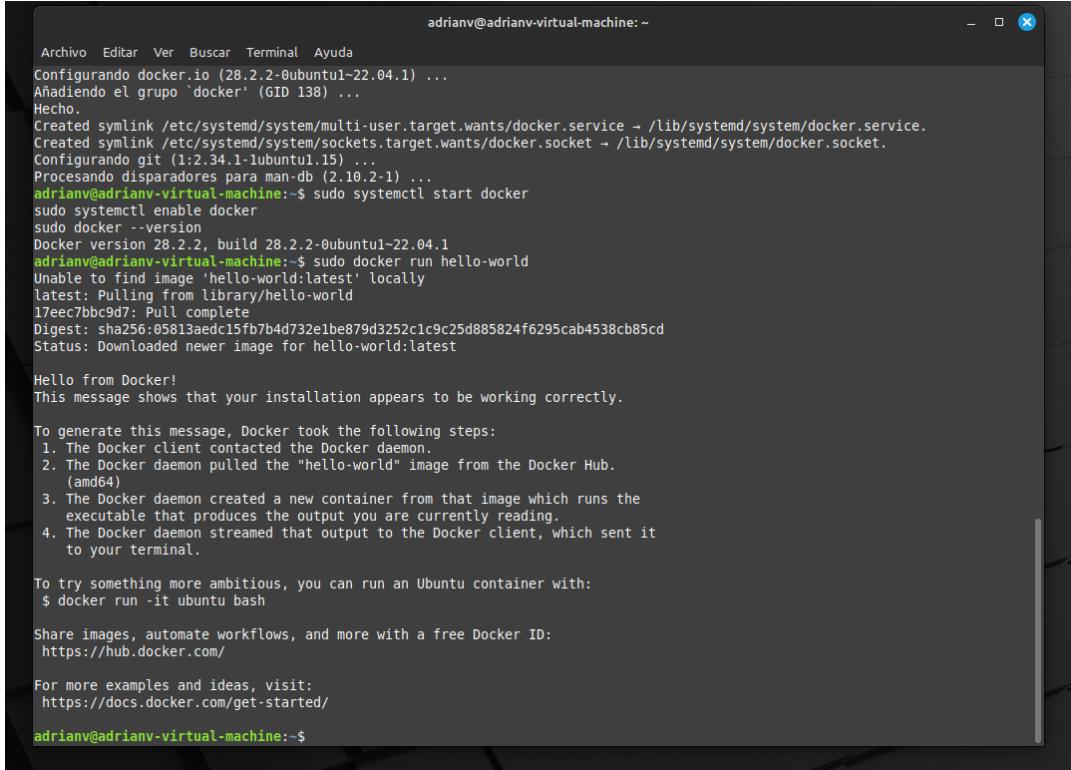
3.- INSTALACIÓN DE DOCKER

Con el comando sudo apt install docker.io comenzamos con la instalación de docker



```
adrianv@adrianv-virtual-machine:~$ sudo apt install docker.io
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  bridge-utils containerd git git-man liberror-perl pigz runc ubuntu-fan
Paquetes sugeridos:
  aufs-tools cgroupfs-mount | cgroup-lite debbootstrap docker-buildx docker-compose-v2 docker-doc rinse git-daemon-run
  | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
Se instalarán los siguientes paquetes NUEVOS:
  bridge-utils containerd docker.io git git-man liberror-perl pigz runc ubuntu-fan
0 actualizados, 9 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 80,1 MB de archivos.
Se utilizarán 309 MB de espacio de disco adicional después de esta operación.
Desea continuar? [S/n] s
Des:1 http://archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63,6 kB]
Des:2 http://archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils amd64 1.7-1ubuntu3 [34,4 kB]
Des:3 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 runc amd64 1.3.3-0ubuntu1-22.04.3 [8,857 kB]
Des:4 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 containerd amd64 1.7.28-0ubuntu1-22.04.1 [38,5 MB]
Des:5 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 docker.io amd64 28.2.2-0ubuntu1-22.04.1 [28,4 MB]
Des:6 http://archive.ubuntu.com/ubuntu jammy/main amd64 liberror-perl all 0.17029-1 [26,5 kB]
Des:7 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 git-man all 1:2.34.1-1ubuntu1.15 [955 kB]
Des:8 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 git amd64 1:2.34.1-1ubuntu1.15 [3.166 kB]
Des:9 http://archive.ubuntu.com/ubuntu jammy/universe amd64 ubuntu-fan all 0.12.16 [35,2 kB]
Descargados 80,1 MB en 9s (8.736 kB/s)
Preconfigurando paquetes ...
Seleccionando el paquete pigz previamente no seleccionado.
(Leyendo la base de datos ... 563517 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../0-pigz_2.6-1_amd64.deb ...
Desempaquetando pigz (2.6-1) ...
Seleccionando el paquete bridge-utils previamente no seleccionado.
Preparando para desempaquetar .../1-bridge-utils_1.7-1ubuntu3_amd64.deb ...
Desempaquetando bridge-utils (1.7-1ubuntu3) ...
Seleccionando el paquete runc previamente no seleccionado.
Preparando para desempaquetar .../2-runc_1.3.3-0ubuntu1-22.04.3_amd64.deb ...
Desempaquetando runc (1.3.3-0ubuntu1-22.04.3) ...
Seleccionando el paquete containerd previamente no seleccionado.
Preparando para desempaquetar .../3-containerd_1.7.28-0ubuntu1-22.04.1_amd64.deb ...
Desempaquetando containerd (1.7.28-0ubuntu1-22.04.1) ...
Seleccionando el paquete docker.io previamente no seleccionado.
```

Una vez instalada el contenedor Docker procedemos a probar su funcionamiento



```
adrianv@adrianv-virtual-machine:~$ sudo apt install docker.io
Configurando docker.io (28.2.2-0ubuntu1-22.04.1) ...
Añadiendo el grupo `docker` (GID 138) ...
Hecho.
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Configurando git (1:2.34.1-1ubuntu1.15) ...
Procesando disparadores para man-db (2.10.2-1) ...
adrianv@adrianv-virtual-machine:~$ sudo systemctl start docker
sudo systemctl enable docker
sudo docker --version
Docker version 28.2.2, build 28.2.2-0ubuntu1-22.04.1
adrianv@adrianv-virtual-machine:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
17eeccbbc9d7: Pull complete
Digest: sha256:05813aedc15fb7b4d732e1be879d3252c1c9c25d885824f6295cab4538cb85cd
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.

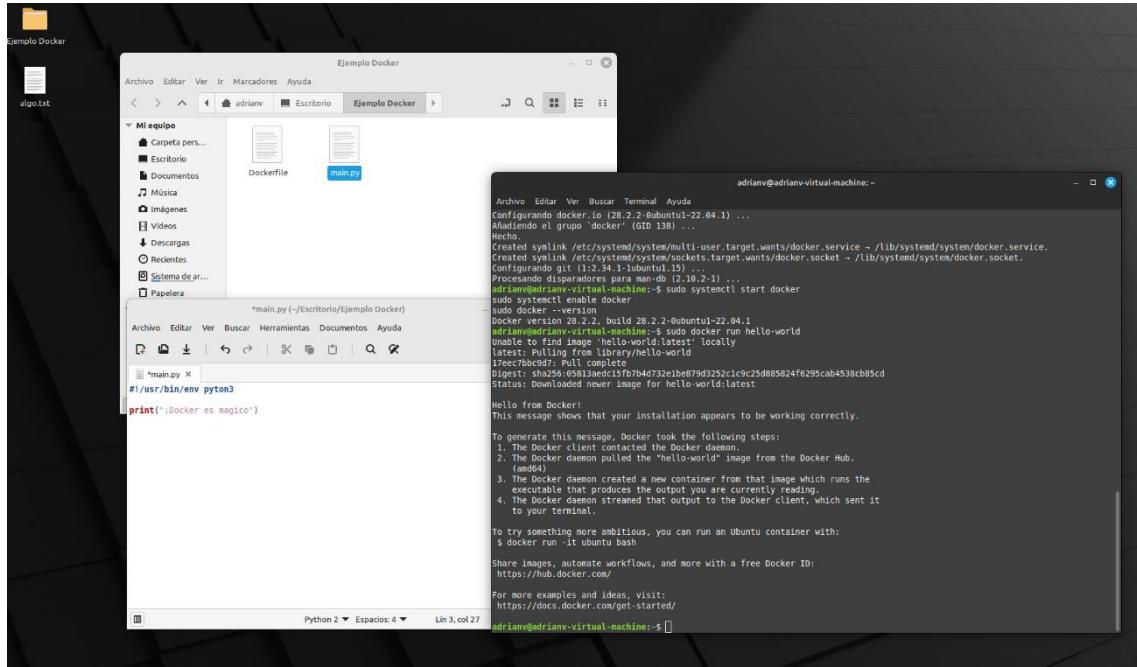
To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

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

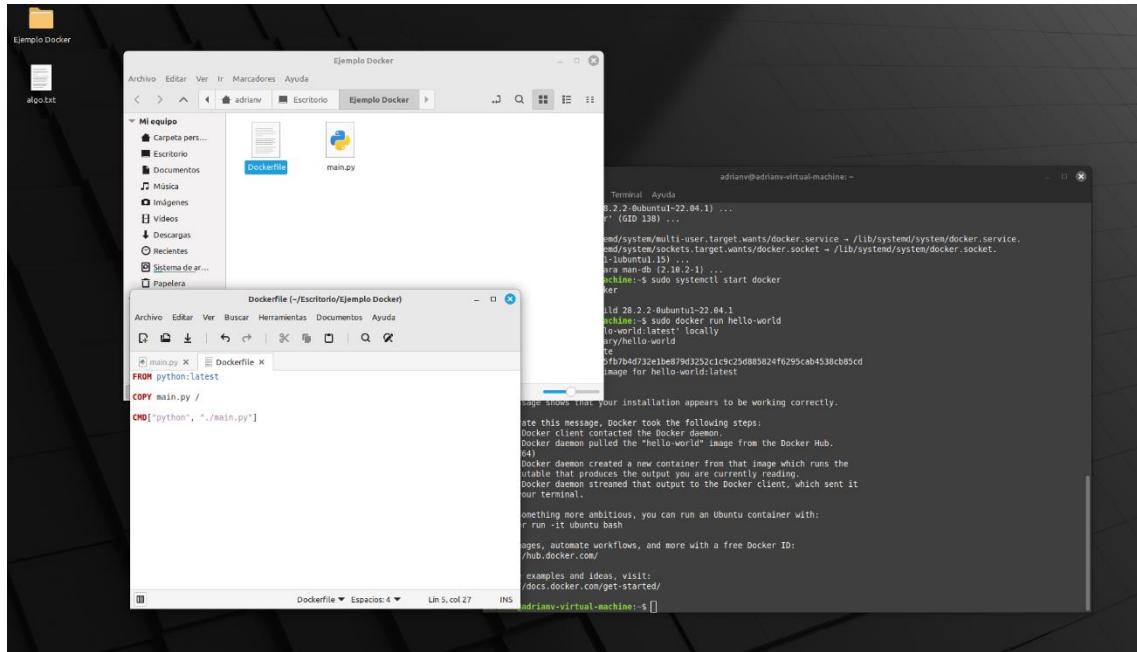
For more examples and ideas, visit:
https://docs.docker.com/get-started/
adrianv@adrianv-virtual-machine:~$
```

4.-CREANDO UN PROYECTO

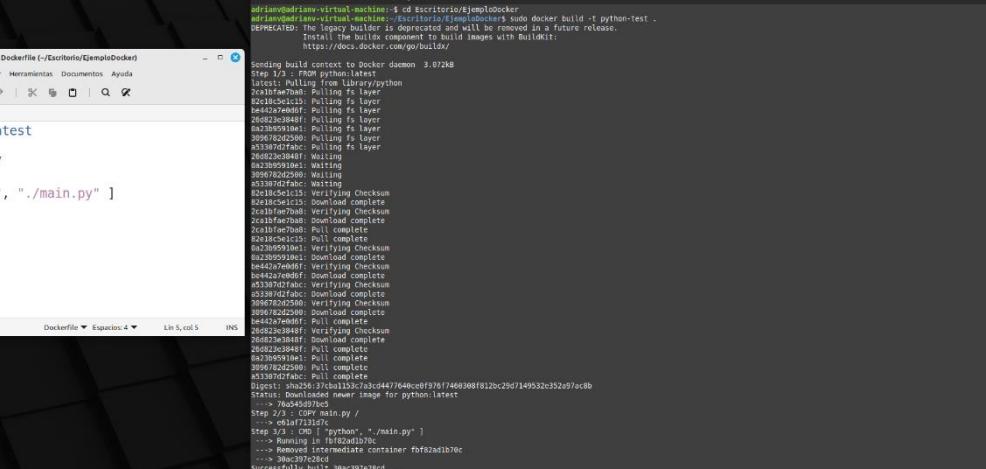
Ahora comenzamos creando una carpeta y a su vez dos archivos dentro como los que se ven en la imagen



Después daremos forma a nuestro archivo dockerfile para que nuestra imagen quede lista



Una vez ya este todo el Docker file listo crearemos la imagen que contendrá nuestra aplicación, iremos al lugar donde hemos creado los archivos y ejecutaremos el comando de la imagen



The screenshot shows a terminal window with the following text:

```
adrian@adrianvm-virtual-machine: ~/Escritorio/EjemploDocker
```

```
Archivo Editor Ver Buscar Terminal Ansible
```

```
adrian@adrianvm-virtual-machine: ~/Escritorio/EjemploDocker
```

```
adrian@adrianvm-virtual-machine: ~/Escritorio/EjemploDocker$ sudo docker build -t python-test .
```

```
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
```

```
INFO: Using the buildx context to build images with BuildKit.
```

```
INFO: The buildx context is configured to use go-buildkit.
```

```
Step 1/3 : FROM python:latest
```

```
Step 1/3 : FROM python:latest
```

```
latest: Pulling from library/python
```

```
260976242300: Pulling fs layer
```

```
b20185c51c15: Pulling fs layer
```

```
b6d42e7e0bf6: Pulling fs layer
```

```
260923c848f1: Pulling fs layer
```

```
260923c848f1: Pulling fs layer
```

```
305978e202300: Pulling fs layer
```

```
a5139705f7abc: Pulling fs layer
```

```
260923c848f1: Pulling fs layer
```

```
a8230595910e: Waiting
```

```
305978e202300: Waiting
```

```
260923c848f1: Waiting
```

```
260923c848f1: Pulling fs layer
```

```
b20185c51c15: Verifying Checksum
```

```
b20185c51c15: Download complete
```

```
260923c848f1: Verifying Checksum
```

```
c2c1bfae0e08ab: Download complete
```

```
c2c1bfae0e08ab: Pull complete
```

```
260923c848f1: Pull complete
```

```
0a7095910e: Verifying Checksum
```

```
0a7095910e: Download complete
```

```
0a7095910e: Pull complete
```

```
b6d42e7e0bf6: Download complete
```

```
b6d42e7e0bf6: Pull complete
```

```
a5139705f7abc: Verifying Checksum
```

```
305978e202300: Verifying Checksum
```

```
305978e202300: Download complete
```

```
305978e202300: Pull complete
```

```
260923c848f1: Verifying Checksum
```

```
260923c848f1: Download complete
```

```
260923c848f1: Pull complete
```

```
a8230595910e: Pull complete
```

```
305978e202300: Verifying Checksum
```

```
305978e202300: Download complete
```

```
305978e202300: Pull complete
```

```
adrian@adrianvm-virtual-machine: ~/Escritorio/EjemploDocker$
```

Una vez creada la imagen la cual la llamamos como nombre phyton-test, pasaremos a lanzarla para ver que contiene.

```
> 30ac397e28cd
Successfully built 30ac397e28cd
Successfully tagged python-test:latest
adrianv@adrianv-virtual-machine:~/Escritorio/EjemploDocker$ sudo docker run python-test
¡Docker es magico
adrianv@adrianv-virtual-machine:~/Escritorio/EjemploDocker$ █
```

Y vemos como nos muestra lo que teníamos en el archivo de Python

5.-COMANDOS VARIOS

Ahora veremos unos cuantos comandos que nos serán útiles en cuanto a las imágenes que hemos creado

5.1 LISTAR

Docker image ls

```
Archivo Editar Ver Buscar Terminal Ayuda
adrianv@adrianv-virtual-machine:~/Escritorio/EjemploDocker$ sudo docker image ls
REPOSITORY      TAG          IMAGE ID      CREATED       SIZE
python-test     latest        30ac397e28cd  5 minutes ago  1.12GB
python          latest        76a545d97be5  44 hours ago  1.12GB
hello-world     latest        1b44b5a3e06a  5 months ago  10.1kB
adrianv@adrianv-virtual-machine:~/Escritorio/EjemploDocker$
```

5.2 ELIMINAR IMAGEN

Docker image rm[nombre de la imagen o ID]

```
adrianv@adrianv-virtual-machine:~/Escritorio/EjemploDocker$ sudo docker image rm 76a545d97be5
Error response from daemon: conflict: unable to delete 76a545d97be5 (cannot be forced) - image has dependent child images
adrianv@adrianv-virtual-machine:~/Escritorio/EjemploDocker$ sudo docker image ls
REPOSITORY      TAG      IMAGE ID      CREATED        SIZE
python-test     latest   30ac397e28cd  8 minutes ago  1.12GB
hello-world     latest   1b44b5a3e06a  5 months ago  10.1kB
adrianv@adrianv-virtual-machine:~/Escritorio/EjemploDocker$
```

5.3 LISTAR CONTENEDORES

sudo docker ps-a

sudo docker ps

```
adrianv@adrianv-virtual-machine:~/Escritorio/EjemploDocker$ sudo docker ps -a
CONTAINER ID        IMAGE           COMMAND       CREATED          STATUS          PORTS          NAMES
6b9269801548      python-test     "python ./main.py" 7 minutes ago   Exited (0) 7 minutes ago
d1976565a0c5      hello-world    "/hello"      11 hours ago    Exited (0) 11 hours ago
adrianv@adrianv-virtual-machine:~/Escritorio/EjemploDocker$
```

```
adrianv@adrianv-virtual-machine:~/Escritorio/EjemploDocker$ sudo docker ps
CONTAINER ID        IMAGE           COMMAND       CREATED          STATUS          PORTS          NAMES
adrianv@adrianv-virtual-machine:~/Escritorio/EjemploDocker$
```

5.4 DETENER UN CONTENEDOR ESPECIFICO

Docker stop[NOMBRE O ID]

```
adrianv@adrianv-virtual-machine:~/Escritorio/EjemploDocker$ sudo docker ps -a
CONTAINER ID        IMAGE           COMMAND       CREATED          STATUS          PORTS          NAMES
6b9269801548      python-test     "python ./main.py" 11 minutes ago   Exited (0) 11 minutes ago
d1976565a0c5      hello-world    "/hello"      11 hours ago    Exited (0) 11 hours ago
adrianv@adrianv-virtual-machine:~/Escritorio/EjemploDocker$ sudo docker stop upbeat_swirles
upbeat_swirles
adrianv@adrianv-virtual-machine:~/Escritorio/EjemploDocker$
```

5.5 ELIMINAR TODOS LOS CONTENEDORES DETENIDOS

sudo docker rm\$(sudo docker ps -a -q)

```
adrianv@adrianv-virtual-machine:~/Escritorio/EjemploDocker$ sudo docker rm $(sudo docker ps -a -q)
6b9269801548
d1976565a0c5
adrianv@adrianv-virtual-machine:~/Escritorio/EjemploDocker$
```

5.6 MOSTRAR REGISTROS DE UN CONTENEDOR

Docker logs[nombre del contenedor]

```
adrianv@adrianv-virtual-machine:~/Escritorio/EjemploDocker$ sudo docker run -d --name miprueba pruebadocker
fc45d325200d993401cff96ff2e235b374f3b6e12bb083d427f15228341b7a6
adrianv@adrianv-virtual-machine:~/Escritorio/EjemploDocker$ sudo docker logs miprueba
¡Docker es magico
adrianv@adrianv-virtual-machine:~/Escritorio/EjemploDocker$
```

Aquí en este punto para no copiar el id decidí renombrar el archivo y ponerle un nombre que no sea números y así poder llamarlo directamente por el nombre.