



Arellano Granados Angel Mariano

218123444

Computación Tolerante a Fallas

D06 2023B

Ejemplo utilizando Docker

Introducción

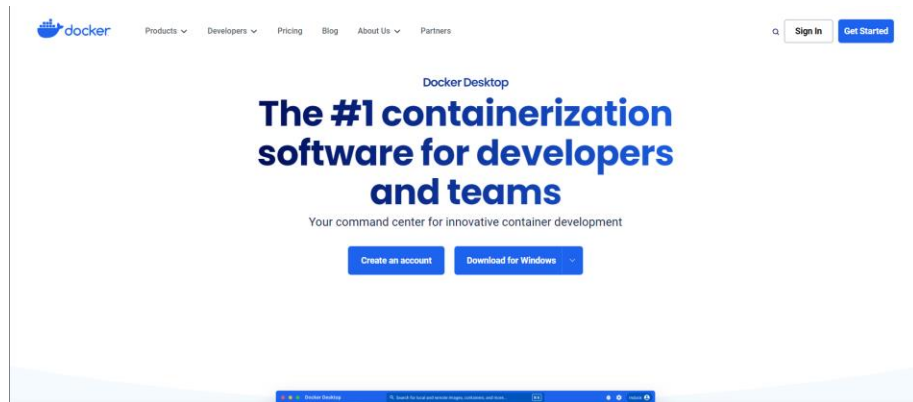
Dentro de este documento se desarrollará un ejemplo de uso simple para la herramienta Docker, abarcando desde la instalación en un sistema operativo Windows 10 hasta la implementación de un pequeño ejemplo.

Docker es una herramienta o plataforma que nos permite probar o instalar aplicaciones de manera aislada de manera rápida y simple, Docker empaqueta varios softwares en diferentes contenedores los cuales incluyen los requisitos para su correcto funcionamiento como las librerías.

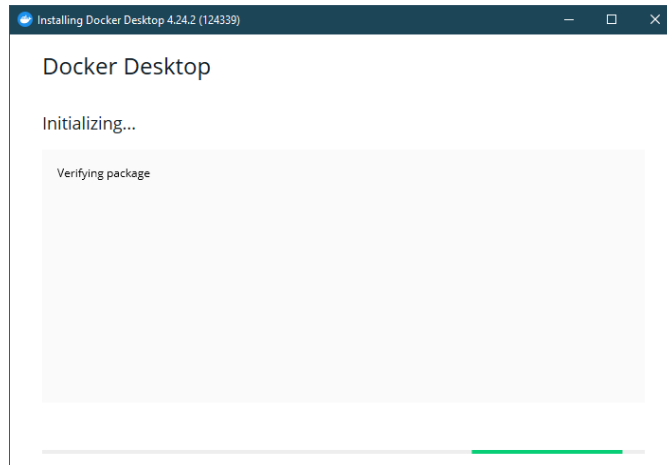
Desarrollo

Instalación:

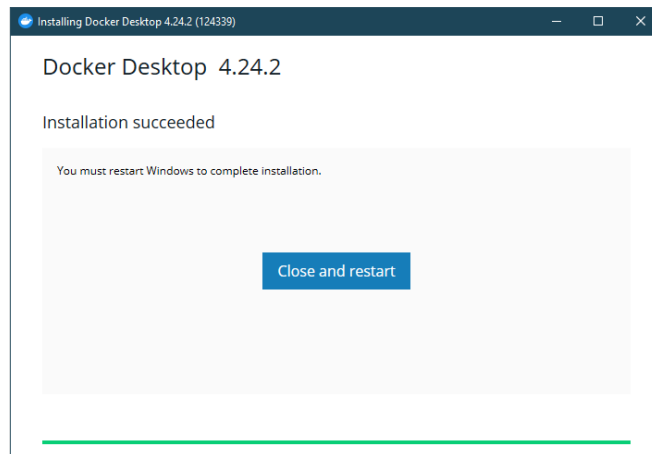
Visitamos la página oficial de Docker y descargamos el instalador para nuestro sistema operativo.



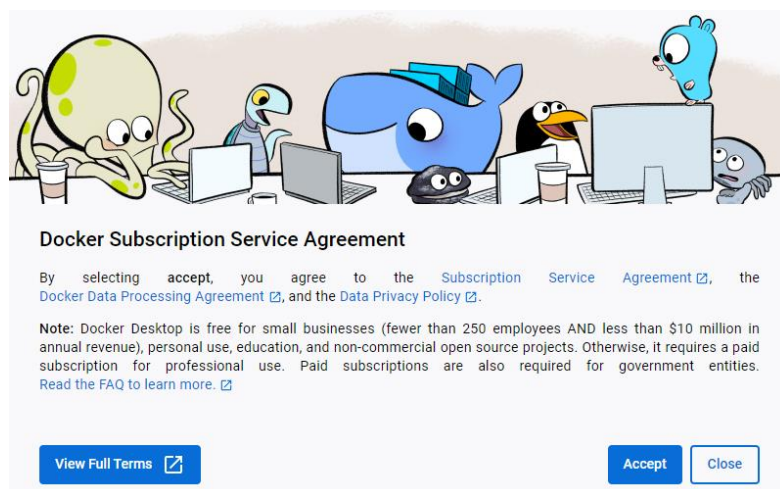
Abrimos el archivo descargado "Docker Desktop Installer.exe" y continuemos con la instalación predeterminada.



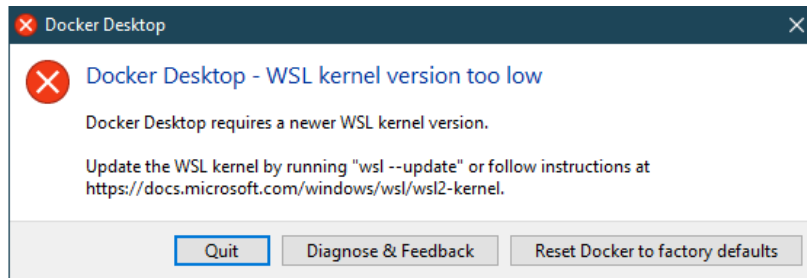
Esperamos a que la instalación termine y reiniciamos nuestro sistema



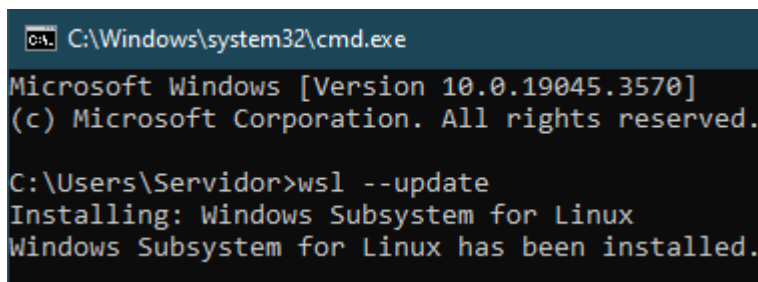
Tras reiniciar nuestro sistema se abrirá de nuevo Docker para aceptar los términos y condiciones



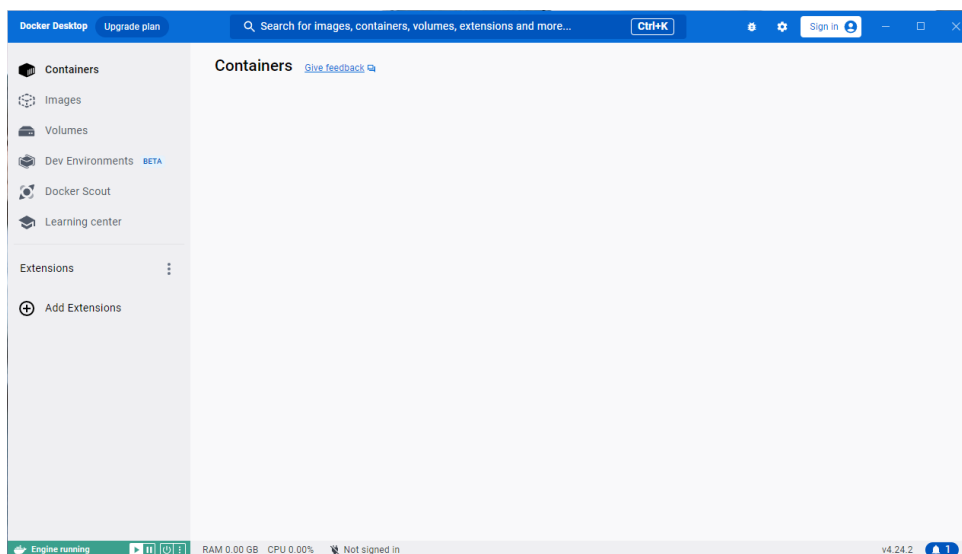
En este punto es probable que nos aparezca el siguiente error



Para solucionarlo tendremos que abrir una consola de Windows y poner el comando "wsl --update" aceptaremos los cambios y esperamos a que se actualice



Salimos del mensaje de error y volvemos a abrir Docker para visualizar la interfaz de usuario



Ahora nos creamos una cuenta de Docker e iniciamos sesión dentro de la aplicación de escritorio



Create your account

Signing up for Docker is fast and free.

☐ Send me occasional product updates and announcements.

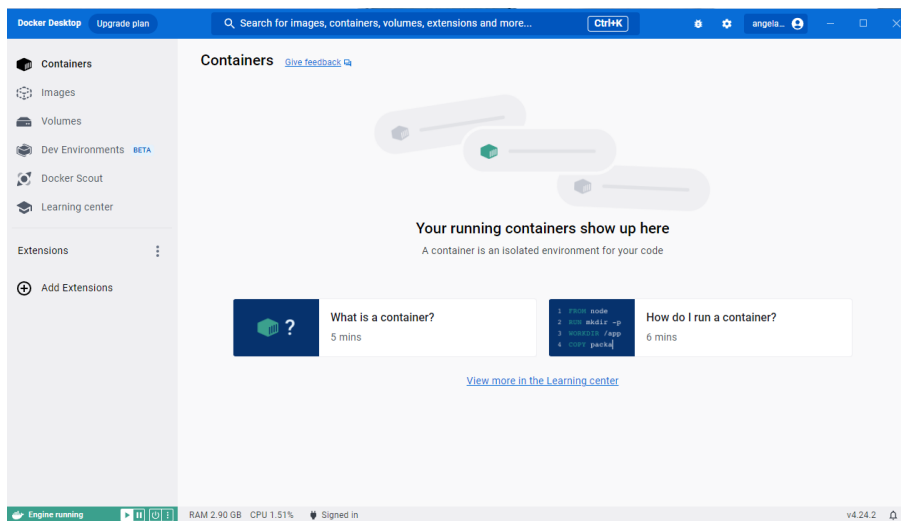
This site is protected by reCAPTCHA and the Google [Privacy Policy](#) and [Terms of Service](#) apply.

Sign up

By creating an account I agree to the [Subscription Service Agreement](#), [Privacy Policy](#), [Data Processing Terms](#).

[Already have an account? Sign in](#)

Y listo ya tenemos Docker listo para ser usador



Ejemplo visto en clase

Comprobamos que no tenemos imágenes de programas ni procesos iniciados en Docker y después instalamos alpine el cual es un versión muy pequeña de Linux.

```

C:\Windows\system32\cmd.exe - docker run -it alpine:3.18.4 sh
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Servidor>docker images
REPOSITORY    TAG        IMAGE ID      CREATED      SIZE

C:\Users\Servidor>docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED      STATUS      PORTS      NAMES

C:\Users\Servidor>docker pull alpine:3.18.4
3.18.4: Pulling from library/alpine
96526aa774ef: Pull complete
Digest: sha256:eece025e432126ce23f223450a0326fbdebde39cdf496a85d8c016293fc851978
Status: Downloaded newer image for alpine:3.18.4
docker.io/library/alpine:3.18.4

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview alpine:3.18.4

```

Abrimos alpine para acceder a una consola de Linux donde actualizamos apk para instalar curl y hacer una prueba con una pagina como Google.

```

C:\Windows\system32\cmd.exe

C:\Users\Servidor>docker run -it alpine:3.18.4 sh
/ # apk update
fetch https://dl-cdn.alpinelinux.org/alpine/v3.18/main/x86_64/AP
fetch https://dl-cdn.alpinelinux.org/alpine/v3.18/community/x86_
v3.18.4-129-gdb5b5ce6661 [https://dl-cdn.alpinelinux.org/alpine/
v3.18.4-130-gea49896cd30 [https://dl-cdn.alpinelinux.org/alpine/
OK: 20073 distinct packages available
/ # apk upgrade
(1/7) Upgrading musl (1.2.4-r1 -> 1.2.4-r2)
(2/7) Upgrading busybox (1.36.1-r2 -> 1.36.1-r4)
Executing busybox-1.36.1-r4.post-upgrade
(3/7) Upgrading busybox-binsh (1.36.1-r2 -> 1.36.1-r4)
(4/7) Upgrading libcrypto3 (3.1.3-r0 -> 3.1.4-r0)
(5/7) Upgrading libssl3 (3.1.3-r0 -> 3.1.4-r0)
(6/7) Upgrading ssl_client (1.36.1-r2 -> 1.36.1-r4)
(7/7) Upgrading musl-utils (1.2.4-r1 -> 1.2.4-r2)
Executing busybox-1.36.1-r4.trigger
OK: 7 MiB in 15 packages
/ # apk upgrade
OK: 7 MiB in 15 packages
/ # curl www.google.com
sh: curl: not found
/ # curl www.google.com
sh: curl: not found
/ # apk add curl
(1/7) Installing ca-certificates (20230506-r0)
(2/7) Installing brotli-libs (1.0.9-r14)
(3/7) Installing libunistring (1.1-r1)
(4/7) Installing libidn2 (2.3.4-r1)
(5/7) Installing nghttp2-libs (1.57.0-r0)
(6/7) Installing libcurl (8.4.0-r0)
(7/7) Installing curl (8.4.0-r0)
Executing busybox-1.36.1-r4.trigger
Executing ca-certificates-20230506-r0.trigger
OK: 12 MiB in 22 packages
/ # curl www.google.com
<!doctype html><html itemscope="" itemtype="http://schema.org/We
rset=UTF-8" http-equiv="Content-Type"><meta content=""/images/bra
rop="image"><title>Google</title><script nonce="LI5MXHibH1XxhgVq

```

Ahora repetimos el proceso, pero esta vez instalamos nginx.

```
C:\Windows\system32\cmd.exe
C:\Users\Servidor>docker pull nginx:1.23
1.23: Pulling from library/nginx
f03b40093957: Pull complete
0972072e0e8a: Pull complete
a85095acb896: Pull complete
d24b987aa74e: Pull complete
6c1a86118ade: Pull complete
9989f7b33228: Pull complete
Digest: sha256:f5747a42e3adcb3168049d63278d7251d91185bb5111d2563d58729a5c9179b0
Status: Downloaded newer image for nginx:1.23
docker.io/library/nginx:1.23

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview nginx:1.23
```

Corremos nginx y en otro cmd revisamos que el proceso si se está ejecutando.

```
C:\Users\Servidor>docker run nginx:1.23
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/10/30 00:10:52 [notice] 1#1: using the "epoll" event method
2023/10/30 00:10:52 [notice] 1#1: nginx/1.23.4
2023/10/30 00:10:52 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2023/10/30 00:10:52 [notice] 1#1: OS: Linux 5.15.90.1-microsoft-standard-WSL2
2023/10/30 00:10:52 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/10/30 00:10:52 [notice] 1#1: start worker processes
2023/10/30 00:10:52 [notice] 1#1: start worker process 29
2023/10/30 00:10:52 [notice] 1#1: start worker process 30
2023/10/30 00:10:52 [notice] 1#1: start worker process 31
2023/10/30 00:10:52 [notice] 1#1: start worker process 32
```

```
Command Prompt
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Servidor>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
315a6dce064d   nginx:1.23  "/docker-entrypoint...." About a minute ago Up About a minute 80/tcp       suspicious_hertz
```

Probamos ahora ejecutar otro nginx pero sin bloquear la terminal para poder visualizarlo y detenerlo.

```
Command Prompt
C:\Users\Servidor>docker run -d nginx:1.23
62b2ca852df23869af049336b4ecd72260d4e0616a789641dd557a1b07d42d8

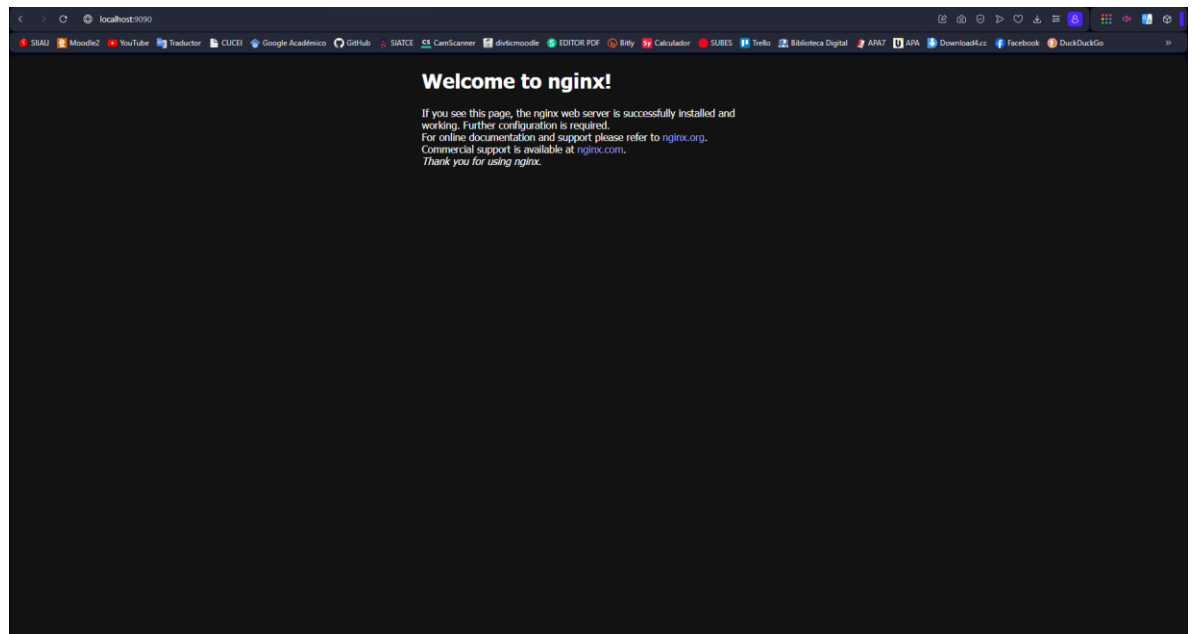
C:\Users\Servidor>docker logs 62b2ca852df23869af049336b4ecd72260d4e0616a789641dd557a1b07d42d8
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/10/30 00:14:02 [notice] 1#1: using the "epoll" event method
2023/10/30 00:14:02 [notice] 1#1: nginx/1.23.4
2023/10/30 00:14:02 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2023/10/30 00:14:02 [notice] 1#1: OS: Linux 5.15.90.1-microsoft-standard-WSL2
2023/10/30 00:14:02 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/10/30 00:14:02 [notice] 1#1: start worker processes
2023/10/30 00:14:02 [notice] 1#1: start worker process 29
2023/10/30 00:14:02 [notice] 1#1: start worker process 30
2023/10/30 00:14:02 [notice] 1#1: start worker process 31
2023/10/30 00:14:02 [notice] 1#1: start worker process 32

C:\Users\Servidor>docker stop 62b2ca852df23869af049336b4ecd72260d4e0616a789641dd557a1b07d42d8
62b2ca852df23869af049336b4ecd72260d4e0616a789641dd557a1b07d42d8
```

Ahora se probará abrirá un puerto donde se correrá un nginx el cual podemos visualizar desde nuestro navegador.

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Servidor>docker run -d -p 9090:80 nginx:1.23
1c20a46371405d79620cb73823aa9b92fe783917760bfd6cce11cd8ec933b569
```

```
C:\Windows\system32\cmd.exe - docker start -i 1c20a4637140

C:\Users\Servidor>docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
1c20a4637140   nginx:1.23 "/docker-entrypoint.     2 minutes ago Up 2 minutes   0.0.0.0:9090->80
/tcp cool_wilson
62b2ca852df2   nginx:1.23 "/docker-entrypoint.     About an hour ago Exited (0) 56 minutes ago
busy_hoover
315a6dce064d   nginx:1.23 "/docker-entrypoint.     About an hour ago Exited (0) 55 minutes ago
suspicious_hertz
9cc6c1cc6054   alpine:3.18.4 "sh"                    2 hours ago    Exited (0) 2 hours ago
peaceful_bhabha

C:\Users\Servidor>docker start -i 1c20a4637140
172.17.0.1 - - [30/Oct/2023:01:16:33 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Safari/537.36 OPR/103.0.0.0" "-"
```

Por último, ahora se creará el mismo proceso pero ahora poniendo le un nombre al proceso

```
C:\Users\Servidor>docker run --name mi-web-app -d -p 9090:80 nginx:1.23
461db0859b4a78af4b155381227f0d0ca09a85010ce56b60bca552355079b3f4

C:\Users\Servidor>docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                NAMES
461db0859b4a   nginx:1.23 "/docker-entrypoint.     6 seconds ago Up 5 seconds   0.0.0.0:9090->80/tcp   mi-web-app
```

Conclusión

Dentro de la actividad se nos presentó el reto de instalar Airflow en Windows el cual constaba de una serie de pasos bastante complejos para solo instalar una herramienta en nuestro sistema, con Docker todo ese proceso de reduciría a un comando en consola o un par de clics en la interfaz gráfica, así

demostrando la efectividad de Docker para simplificar el proceso de instalación y compatibilidad de una infinidad de programas diferentes.