Introducción - Usando el Docker Client

Vamos a ejecutar un comando sencillo, y luego vamos a explicar el flujo de las acciones que se produjeron cuando ese comando se ejecutó.

```
jorge@MacBook-Pro-de-Jorge ~ % docker run hello-world
```

A continuación, presionamos Enter, y vamos a ver muy rápidamente un montón de texto de salir en la terminal:

```
jorge@MacBook-Pro-de-Jorge ~ % docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:ffb13da98453e0f04d33a6eee5bb8e46ee50d08ebe17735fc0779d0349e889e9
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/

jorge@MacBook-Pro-de-Jorge ~ %
```

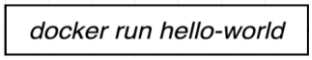
Verá un pequeño mensaje que dice “Hello from Docker!” acompañado de una lista de pasos que acaban de ocurrir cuando se ejecutó el comando.

Nótese también, que aparece un mensaje justo después de dar enter, que dice “*Unable to find image ‘hello-world:latest locally’*”:

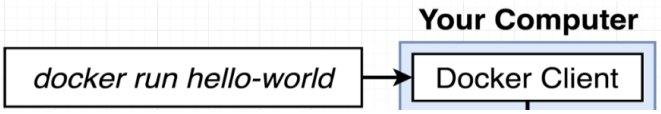
```
jorge@MacBook-Pro-de-Jorge ~ % docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:ffb13da98453e0f04d33a6eee5bb8e46ee50d08ebe17735fc0779d0349e889e9
Status: Downloaded newer image for hello-world:latest

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/

jorge@MacBook-Pro-de-Jorge ~ %
```

Veamos un par de diagramas que van a ayuda a explicar lo que acaba de ocurrir cuando ejecutamos el comando.

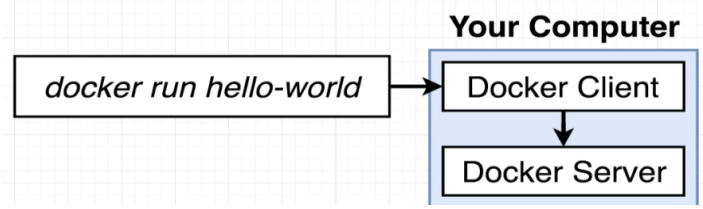
En la terminal, ejecutamos el comando *docker run hello-world:*



Cuando presionamos Enter, se inicia el *Docker Client* o *Docker CLI:*



El *Docker CLI* se encarga de tomar los comandos que ejecutamos y comunicarlos con algo llamado *Docker Server*:

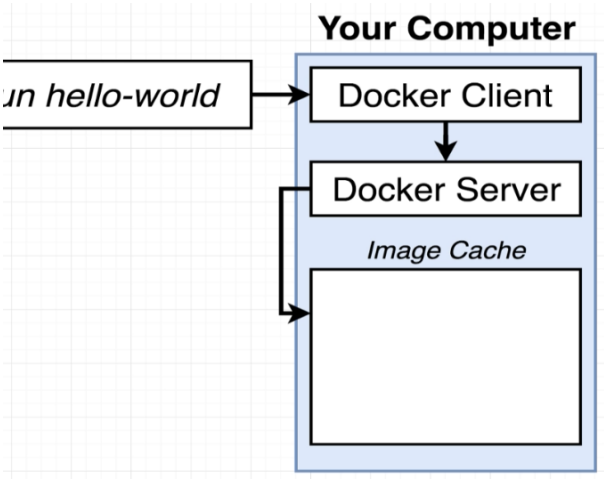


Es este *Docker Server* el que realmente se encarga del trabajo pesado.

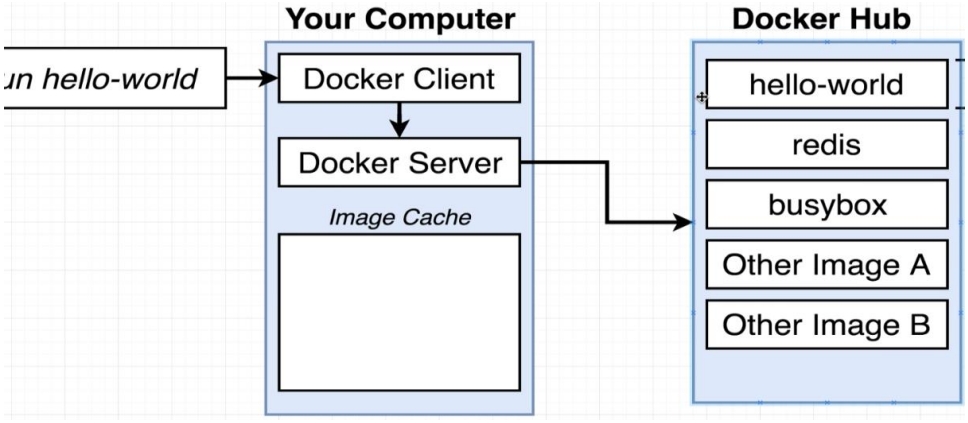
Cuando ejecutamos nuestro comando *docker run hello-world*, significa que queremos iniciar un nuevo *Contenedor* usando la *Imagen* llamado hello-world. La *Imagen* hello-world tiene un pequeño programa dentro de él, cuyo único propósito es imprimir el mensaje que nos apareció al dar Enter.

Ahora, cuando ejecutamos el comando y se envía al Docker Server, una serie de acciones muy rápidamente se produjeron en background:

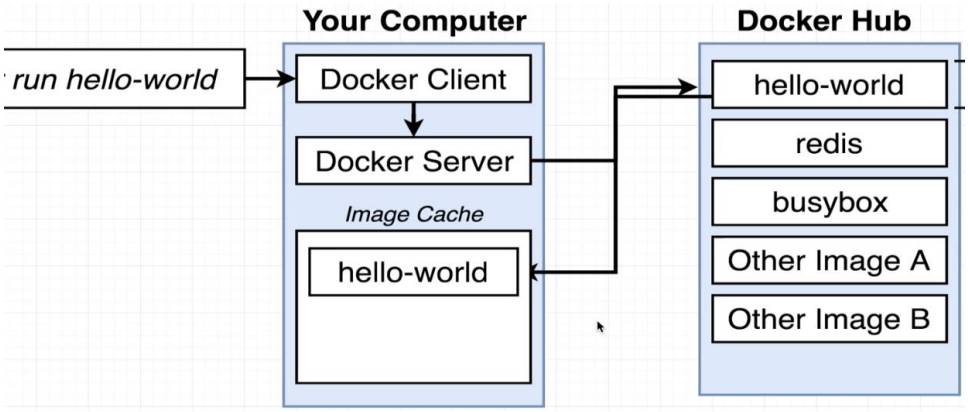
El *Docker Server* identificó que estamos intentando iniciar un nuevo *Contenedor* usando una *Imagen* llamada hello-world. Lo primero que el *Docker Server* hace es comprobar si ya tenemos una copia local, en nuestro equipo local, de la *Imagen* hello-world. Así que el *Docker Server* busca en algo llamado Image Cache:



En nuestro caso, la *Image Cache* está vacía. No tenemos imágenes que se hayan descargado antes. Así que el *Docker Server* procede a invocar un servicio gratuito llamado *Docker Hub*:



El Docker Hub es un repositorio de *Imágenes* públicas gratuitas que puede descargar libremente y ejecutar en su computador. Así que el *Docker Server* se conectó al *Docker Hub* y le solicitó una Imagen llamada hello-world. *Docker Hub* si tiene la Imagen, asi que el *Docker Server* la descarga a su computadora local y lo almacena en el Image Cache:



Ahora podemos volver a ejecutar el comando *docker run hello-world* en cualquier momento sin tener que volver a descargar la imagen desde el *Docker Hub*.

Después de colocar la Imagen en el *Image Cach*e, el *Docker Server* procede a instanciar esa imagen para crear un *Contenedor* y luego ejecutar el programa que está dentro de ese *Contenedor*.

Eso es lo que sucede cuando ejecuta el comando *docker ru*n, se invoca al *Docker Hub* en caso de no tener la imagen en la *Image Cache*, descarga la *Imagen* y luego crea un *Contenedor* con esa *Imagen*.