**Descripción y configuración de Docker**

Docker es una tecnología de código abierto que se utiliza para implementar aplicaciones a través de contenedores. Es relativamente nuevo pero recibe mejoras constantes. Docker es muy popular entre los desarrolladores y se puede instalar sin problemas en tu ordenador en las versiones: Linux, Mac y Windows 10 (home y professional).

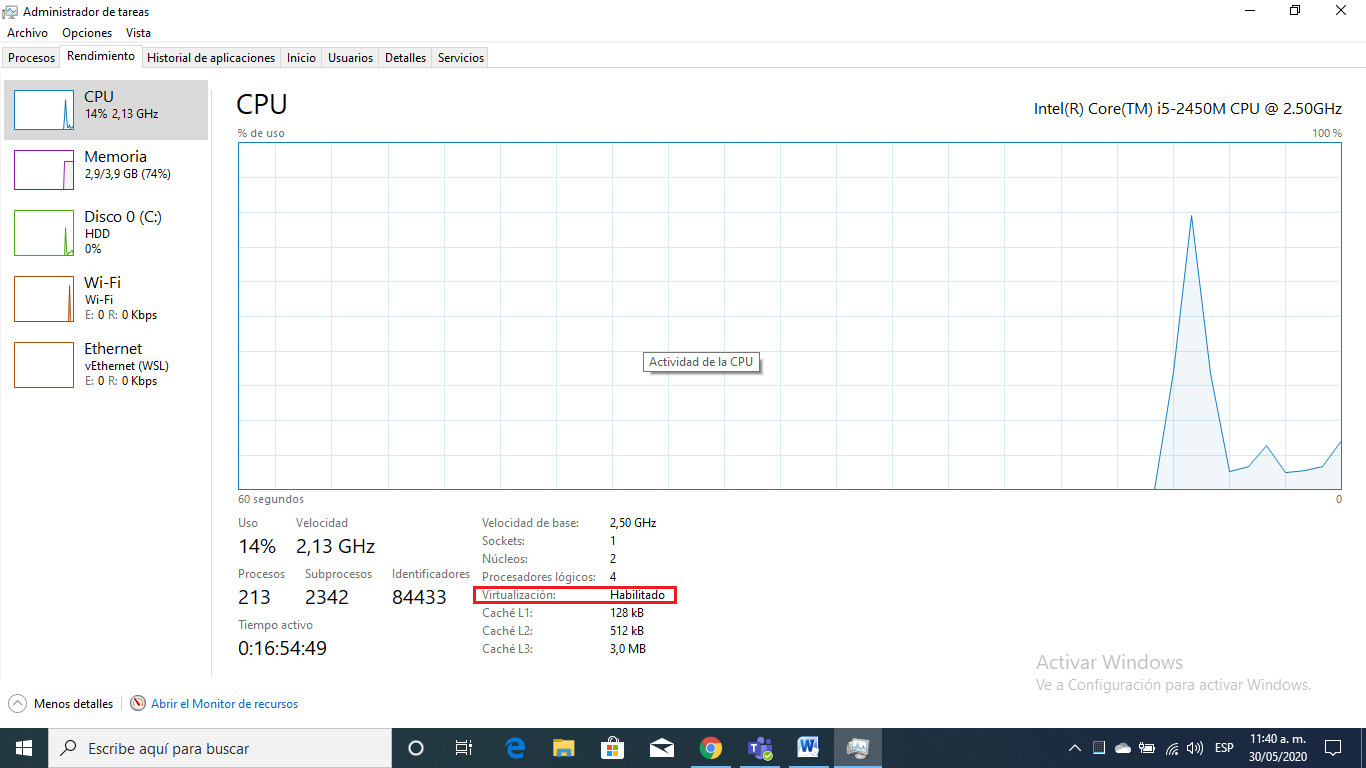
La principal novedad de Docker es que permite empaquetar una aplicación o un conjunto de servicios en contenedores. Un contenedor Docker es la instalación que contiene todas las bibliotecas y componentes necesarios para que una aplicación funcione. Desde un punto de vista práctico, un contenedor es un tipo de máquina virtual reducida que funciona de forma independiente al sistema operativo donde se ejecuta una aplicación o servicio específico.

En este caso describiremos como instalar y dar uso a Docker en Windows 10, si queremos instalar Docker debemos cumplir ciertos requisitos en Windows 10:

* Windows 10 versión Professional o Enteprise.
* Virtualización actividad en nuestro equipo.

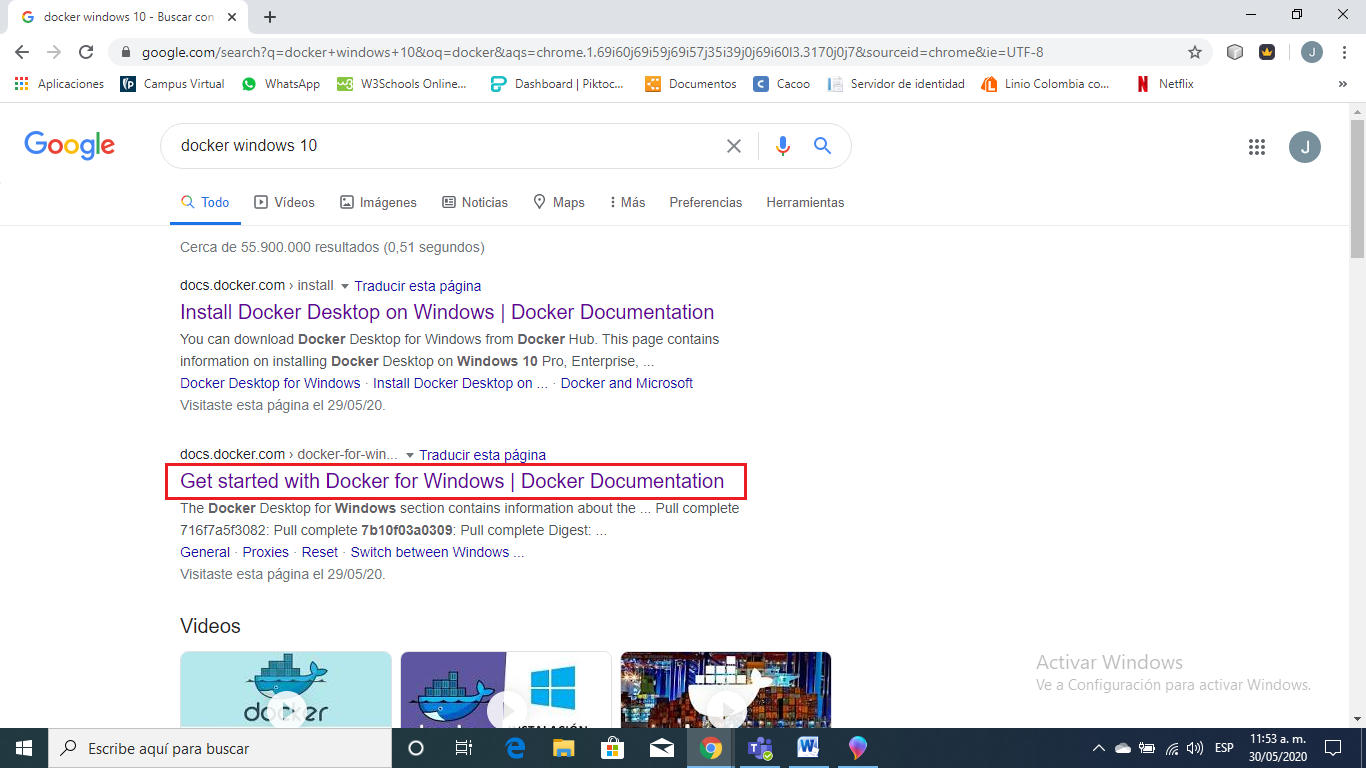
En el caso de no tener la virtualización activada, Docker detectara esta característica en el proceso de instalación, lo que deberemos hacer es ir a la BIOS del equipo y habilitar la virtualización y reiniciar el equipo.

Esta activación la podemos revisar ingresando al administrador de tareas, y dando clic en la pestaña de rendimiento.

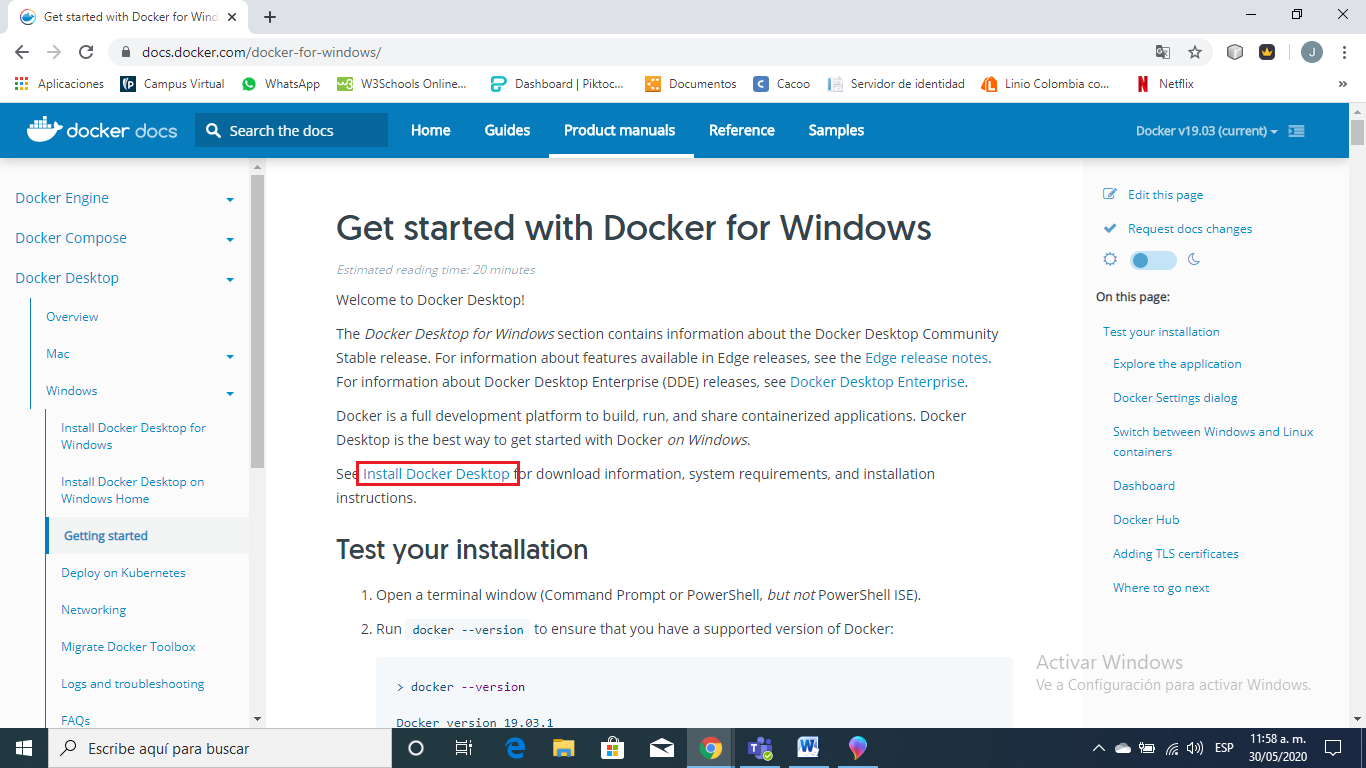


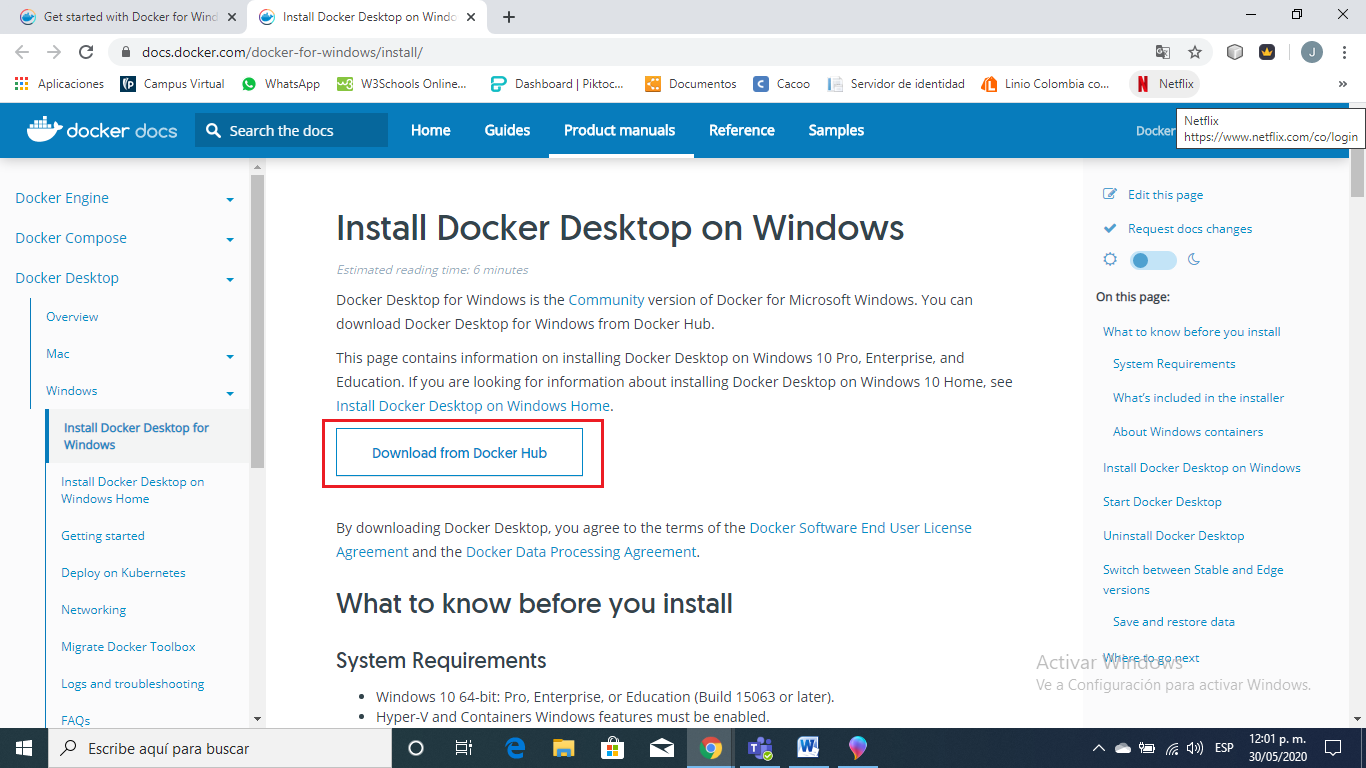
**Instalación de Docker.**

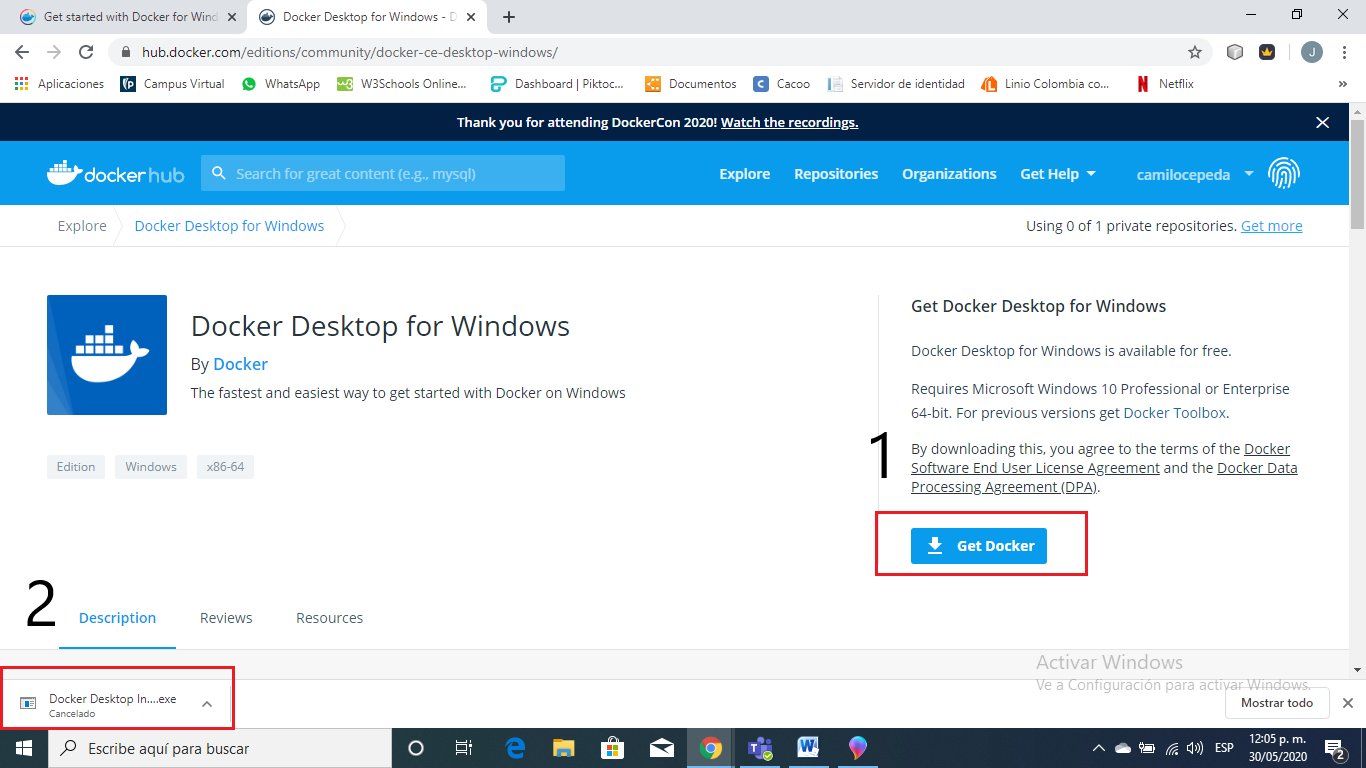
El proceso de instalación de Docker en Windows 10, es bastante simple, lo primero es poner en el buscador Docker Windows 10 el cual nos dará estas opciones y debemos ingresar a la que dice: (Get started with Docker for Windows).



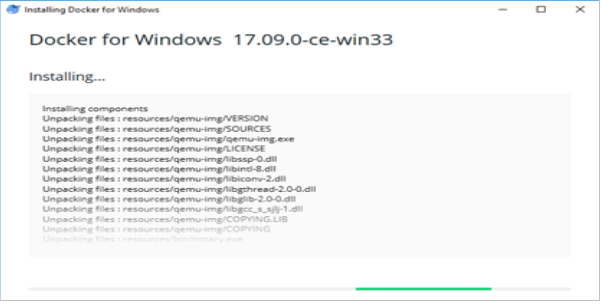
Después damos clic en install Docker desktop el cual nos enviara a la página en la cual podemos descargar el setup de Docker.

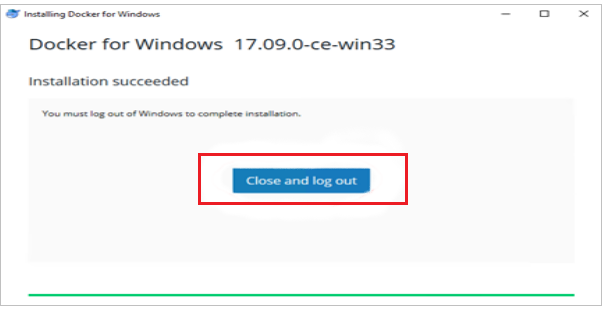




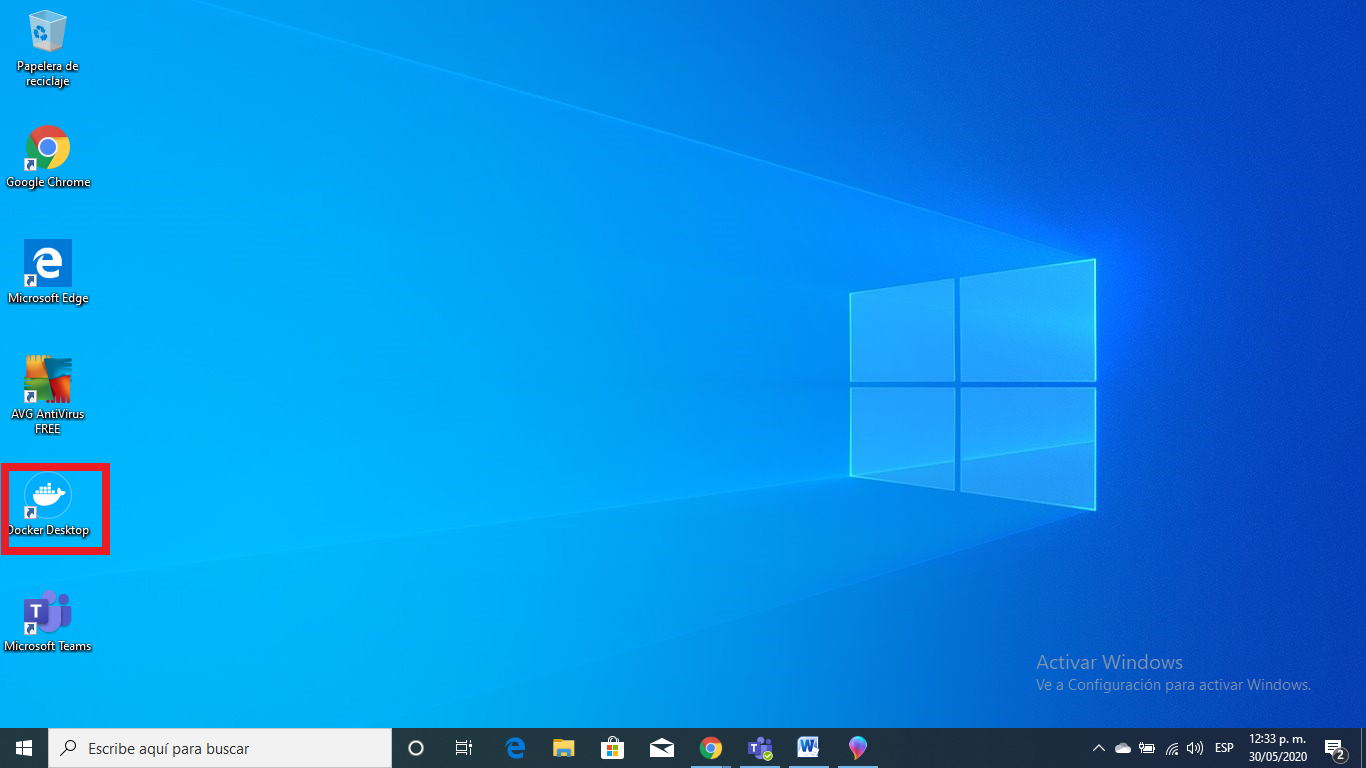


Se lanzara el proceso de instalación, cuando finalice el proceso, aparecerá la ventana de instalación correcta, pulsaremos el botón de (Close and logout) para terminar de configurar Docker en nuestro Windows 10. En algunos casos, esto provocara que se reinicie el sistema Windows para terminar la configuración.





Después de reiniciar nuestro Windows 10 ya tendremos Docker instalado en nuestro equipo.

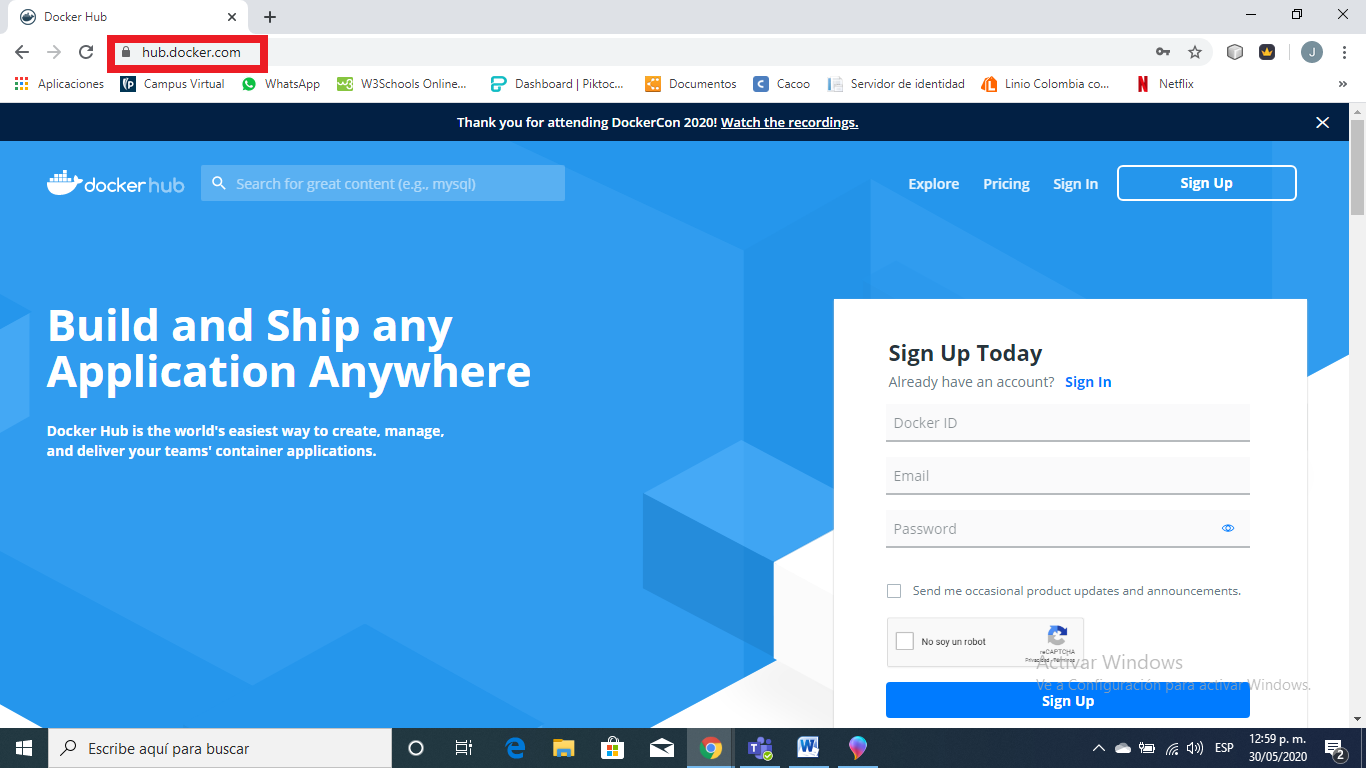


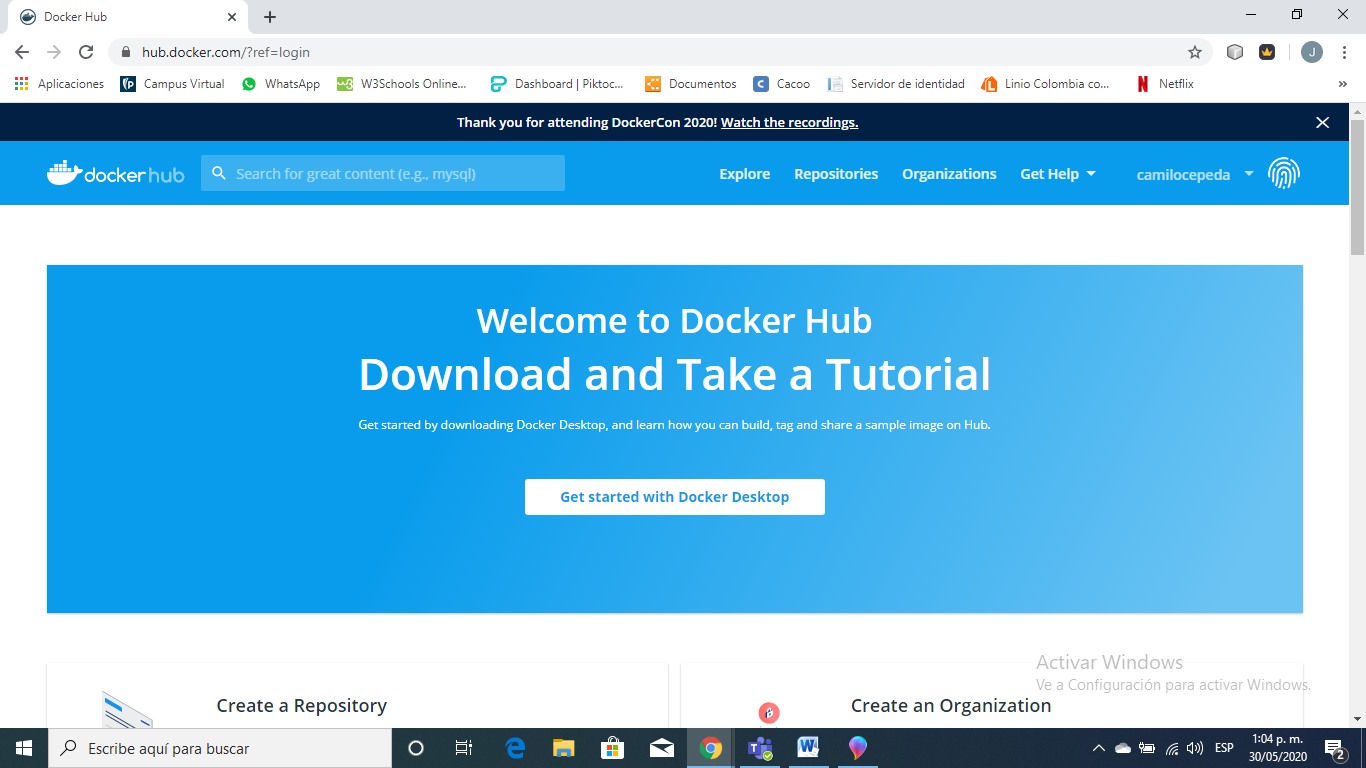
**Descripción y comprobación de imágenes y contenedores.**

Para trabajar con Docker y familiarizarnos con la herramienta, podemos usar la consola (CMD) o PowerShell. En mi caso voy a utilizar PowerShell para empezar a usar Docker.

Aunque no es obligatorio, se recomienda crear una cuneta en Docker Hub para no tener problemas a la hora de descargar imágenes, ya que, algunas nos pedirán nuestras credenciales para descargarse en nuestro equipo.

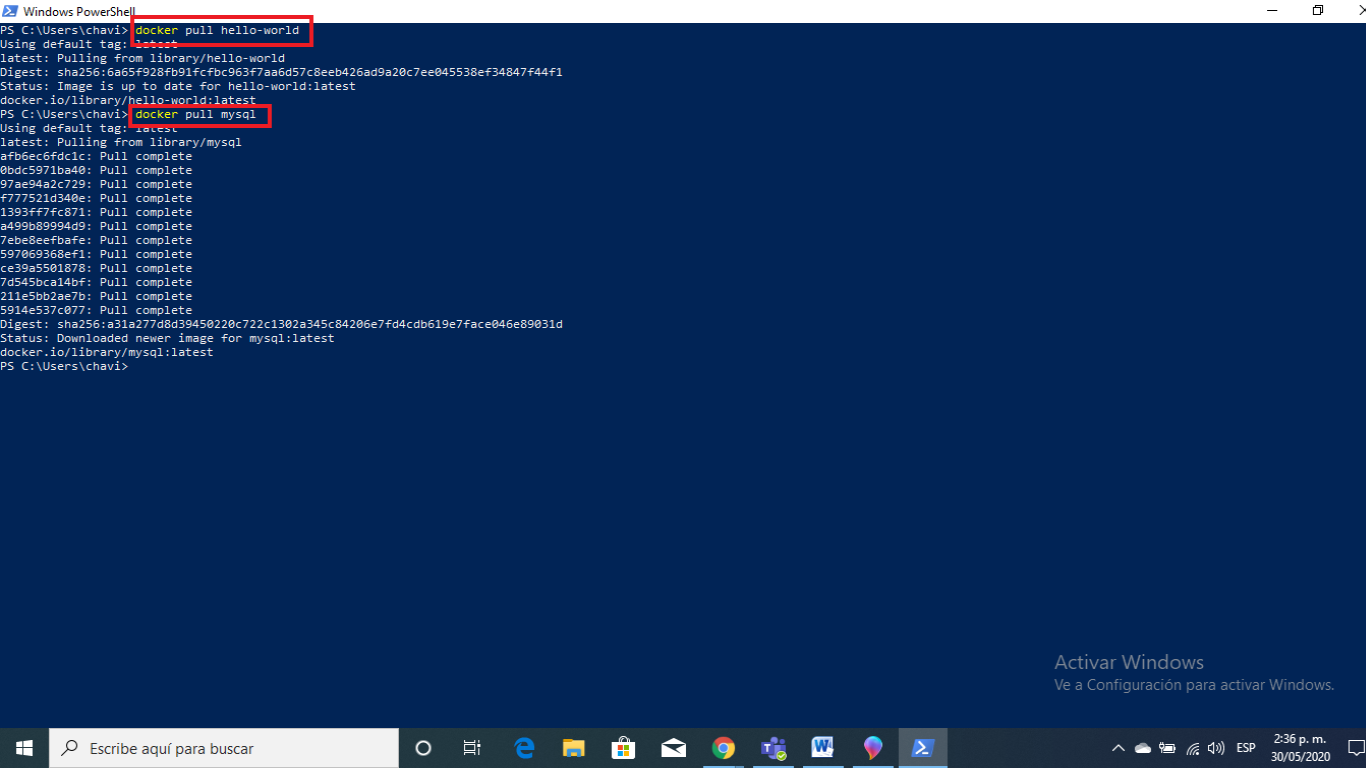
El proceso de registro es muy simple solo se debe ir a la web de Docker Hub, nos registramos y debemos activar la cuenta en el enlace que llega a nuestro correo electrónico.



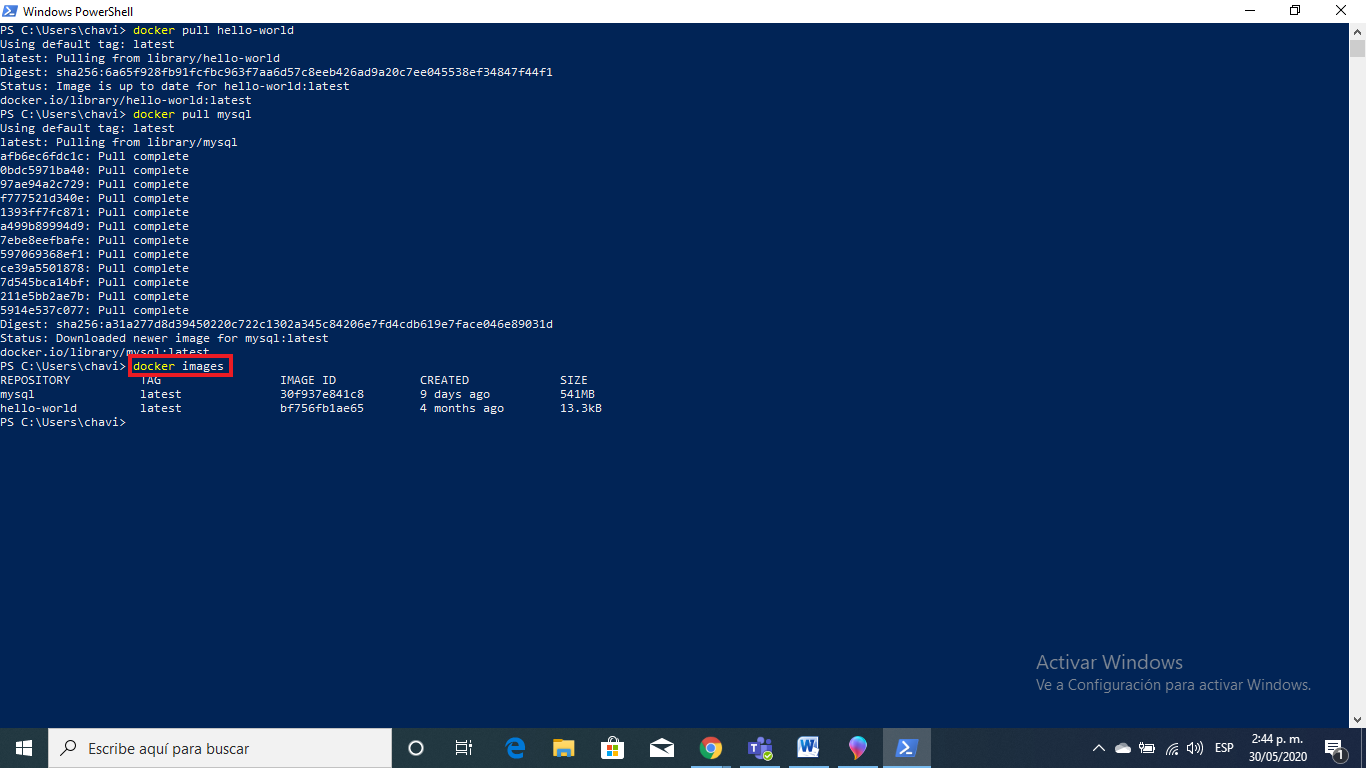


Después de hacer esto ya podemos empezar a trabajar con Docker, lo primero será la creación de imágenes, **Las imágenes en Docker** son plantillas (que incluyen un aplicación, los binarios y las librerías necesarias) que se utilizan para construir contenedores en Docker. En este caso crearemos las imágenes de **hello-world** que es las más conocida y también la imagen de **mysql** la cual es un sistema de gestión de base de datos.

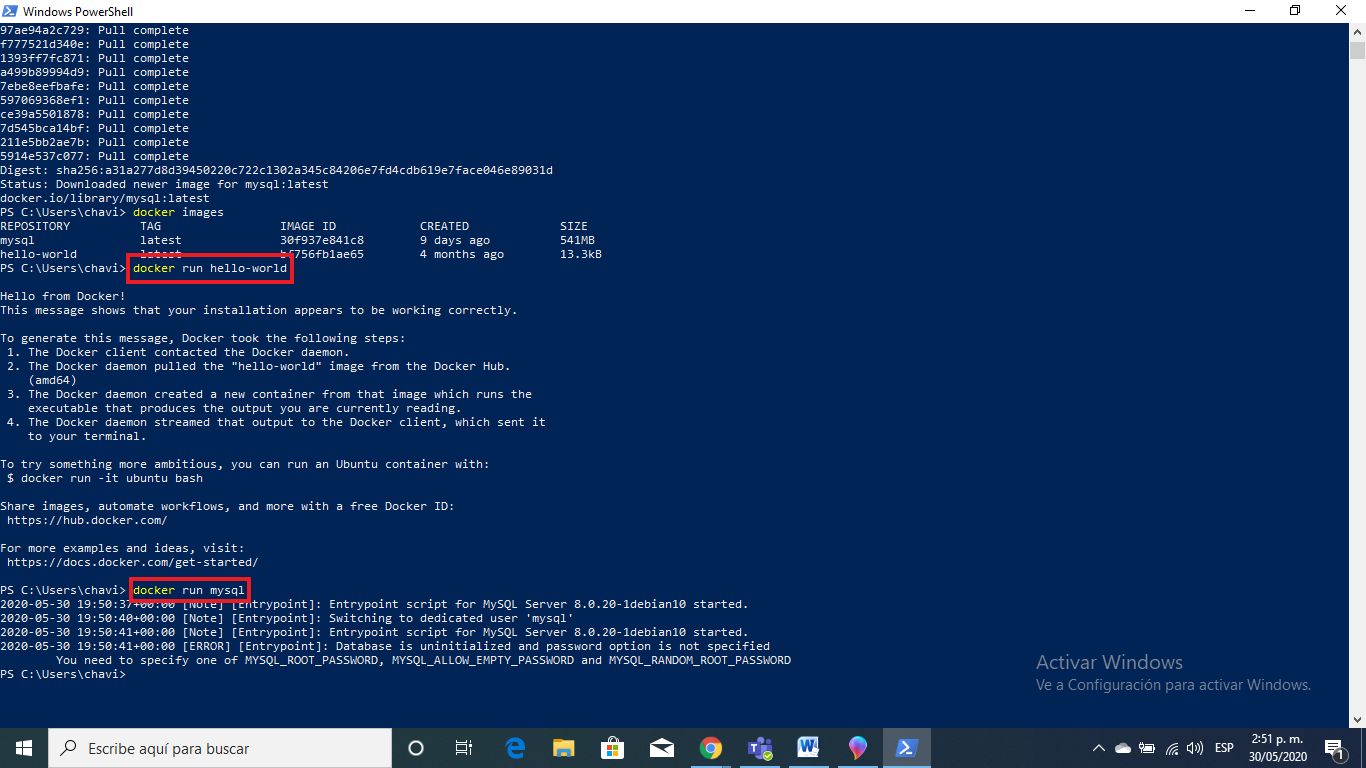
Para descargar una imagen debemos ejecutar el comando (**Docker pull**)y el nombre de la imagen en esta caso sería Docker pull hello-world y Docker pull mysql, con esto descargaremos las imágenes.



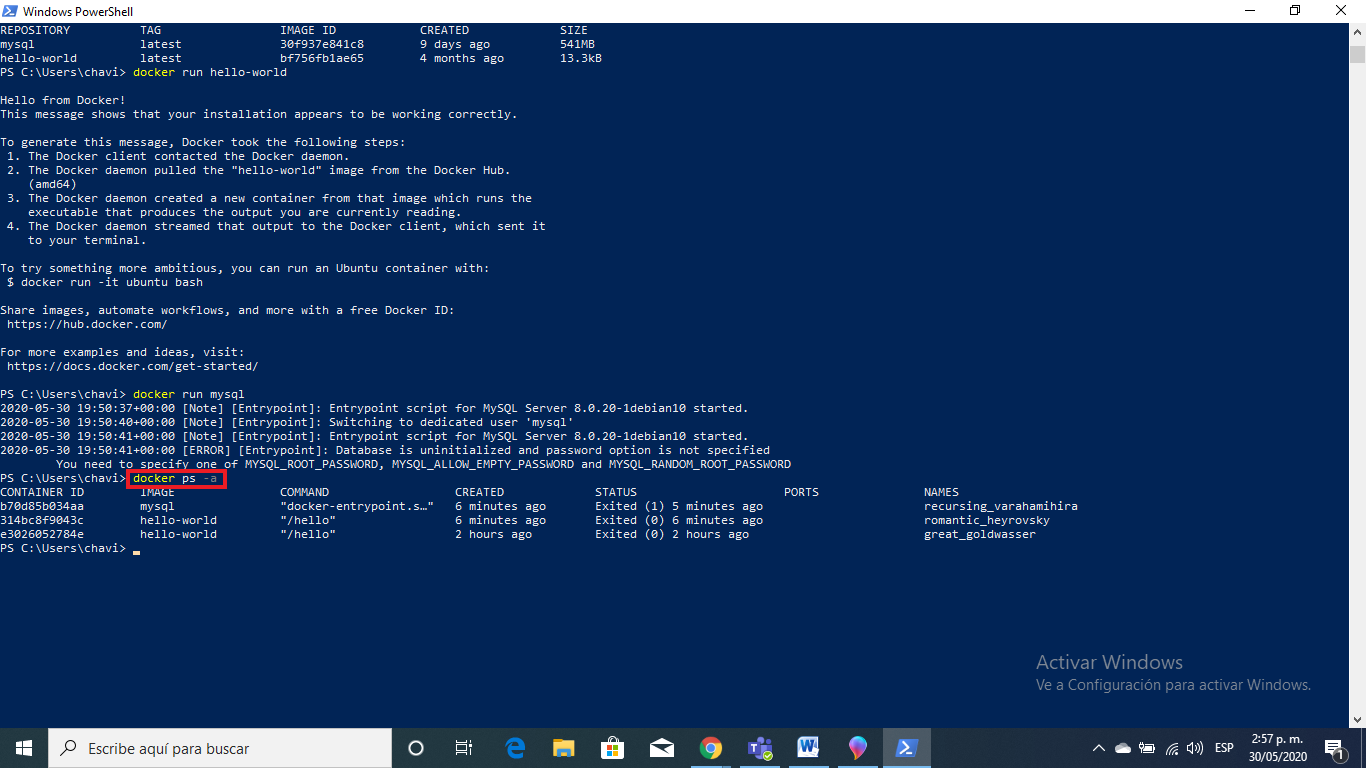
Para saber dónde quedaron nuestras imágenes ejecutamos el comando **Docker images** el cual nos muestra un listado de las imágenes que tenemos con su ID, cuando fueron creadas y cuánto pesa.



Luego necesitamos crear los contenedores para las imágenes, los contenedores se crean ejecutando el comando (**Docker run**) y el nombre de la imagen, en este caso Docker run hello-world y Docker run mysql.



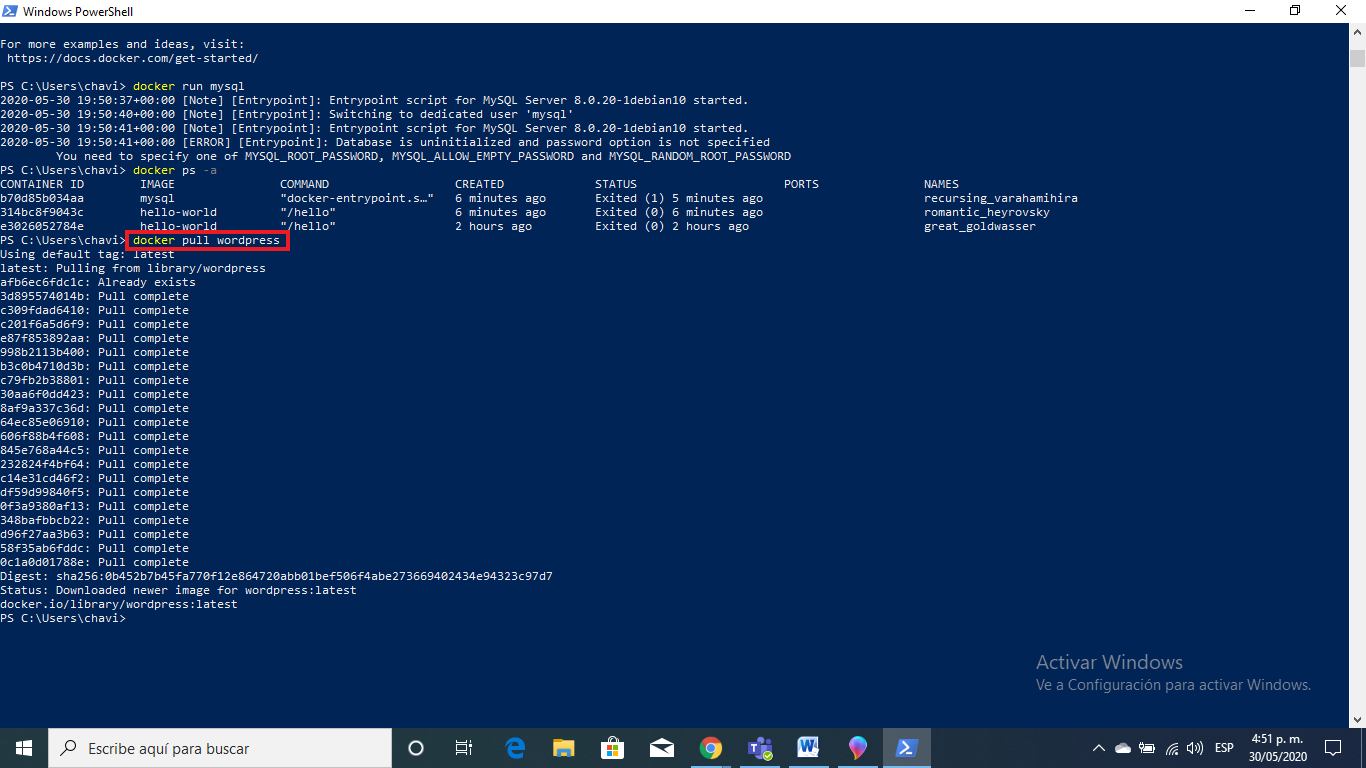
Para ver nuestros contenedores ejecutamos el comando (**docker ps -a**).



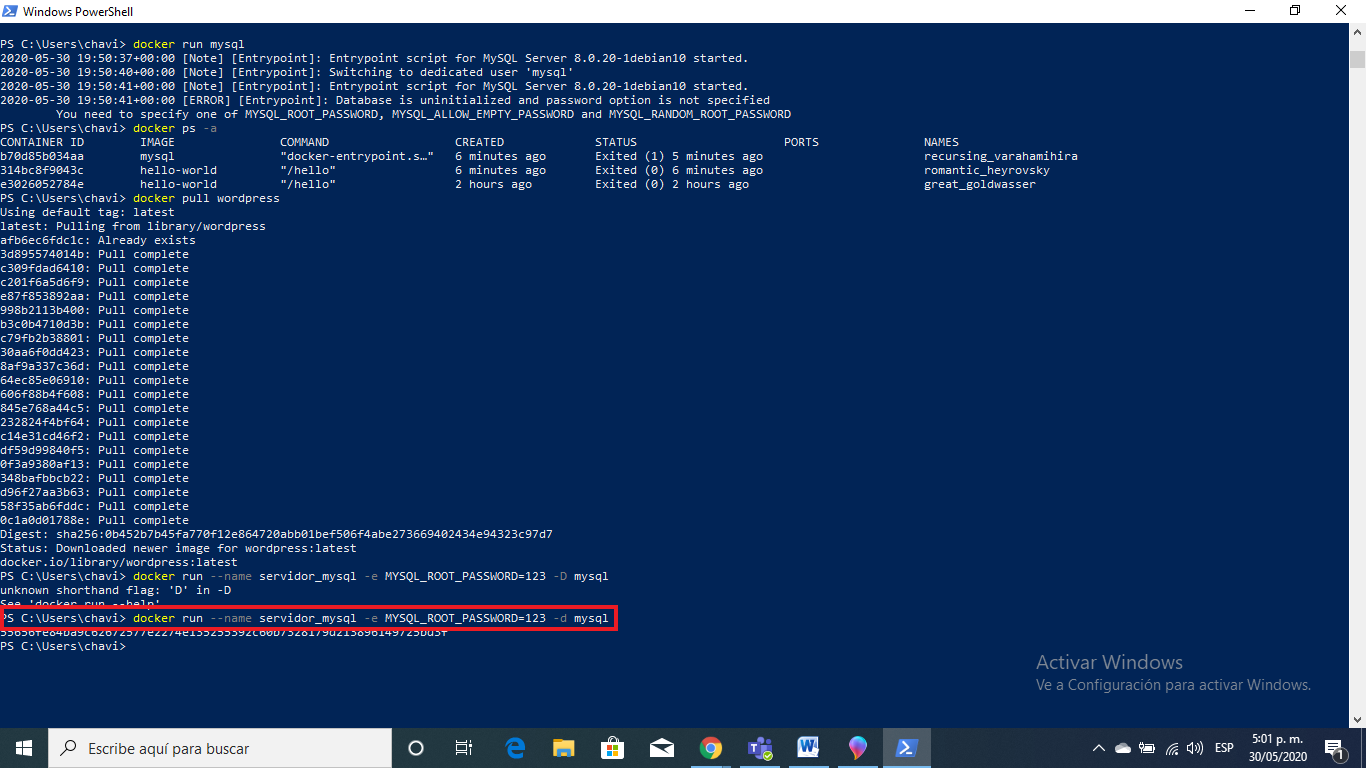
**Comunicación de dos contenedores**

Para el último caso vamos a comunicar dos contenedores entre sí, más concretamente vamos a crear un contenedor con un servidor web con wordpress que vamos a comunicar con otro contenedor con un servidor de base de datos mysql.

Lo primero es crear la imagen de wordpress.



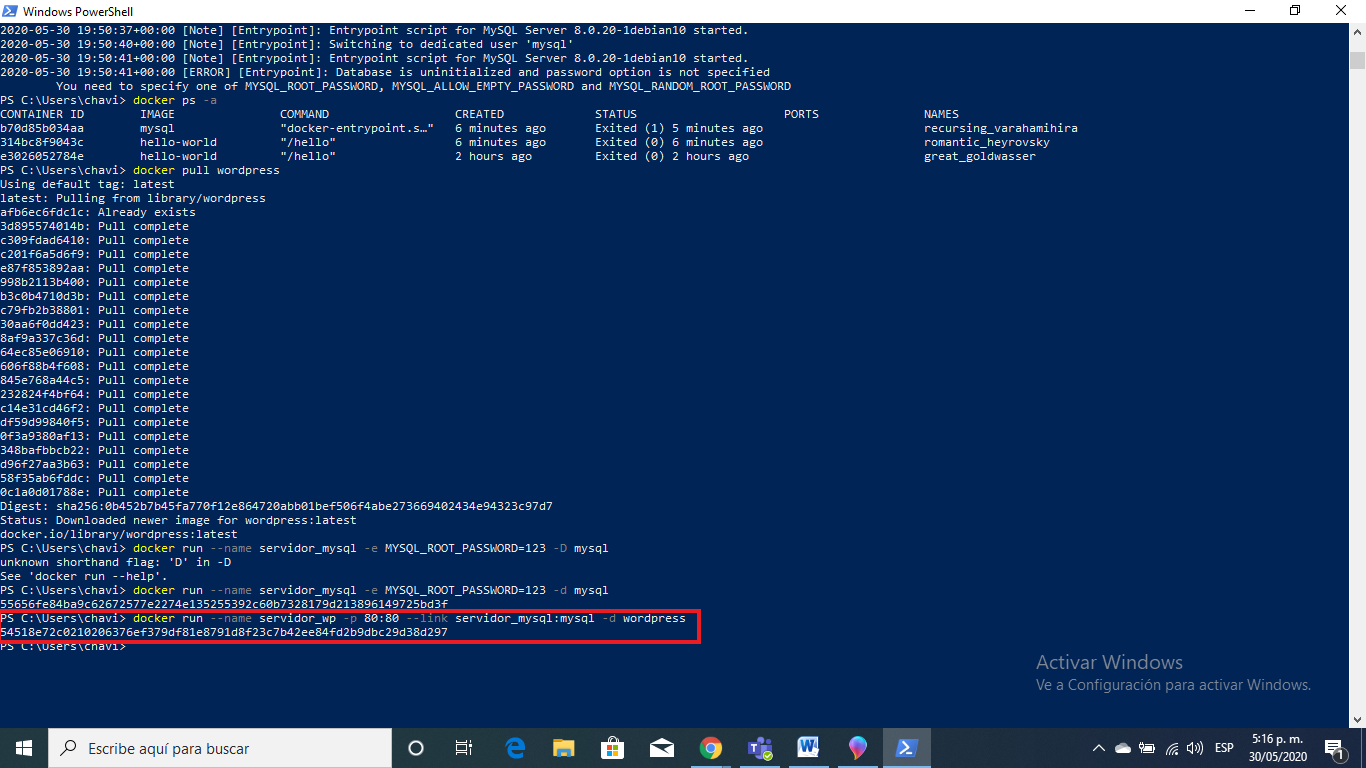
Docker nos permite un mecanismo de enlace entre contenedores. Posibilitando enviar información de forma segura entre ellos y pudiendo compartir información entre ellos, por ejemplo las variables de entorno. Para establecer la asociación entre contenedores es necesario usar el nombre con el que creamos el contenedor, el nombre sirve como punto de referencia para enlazar con otro contenedor.



En este caso solo hemos indicado la variable de entorno **MYSQL\_ROOT\_PASSWORD,** que es obligatoria, indicando la contraseña del usuario root.

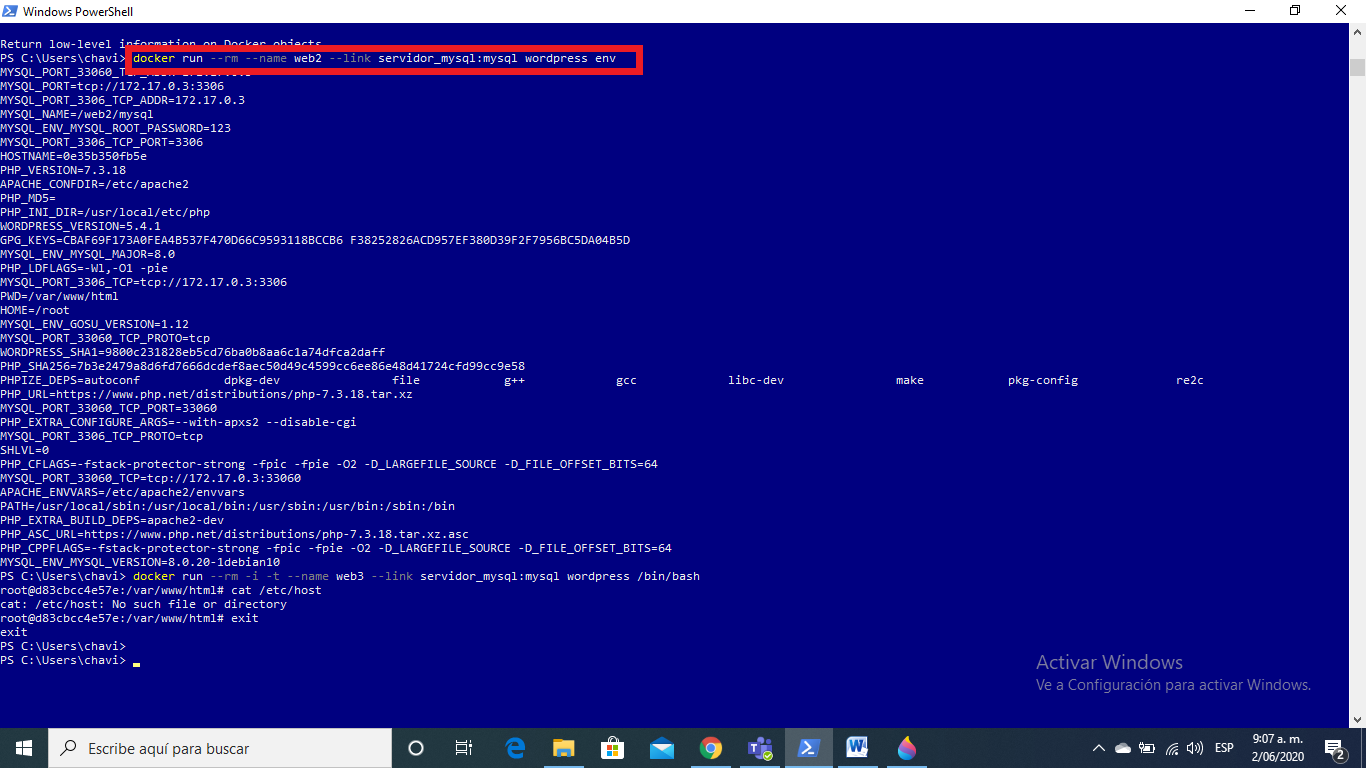
Lo siguiente que vamos hacer es crear un contenedor nuevo, con su respectivo nombre, con el servidor web a partir de la imagen de wordpress, enlazado con el contenedor anterior.

Para realizar la asociación entre contenedores utilice el parámetro **–link**, donde se indica el nombre del contenedor enlazado y un alias por el que nos podemos referir a él como también un puerto que por defecto es 80:80, comandos como **–d** indica que el contenedor corre en segundo plano, **-p** que indica el puerto.



Podemos comprobar la conexión de los contenedores con el siguiente comando Docker run –rm –name web2 –link servidor\_mysql:mysql wordpress env, donde por cada asociación de contenedores, Docker crea una serie de variables de entorno, en este caso, en el contenedor servidor\_wp, se crearan variables, donde se utiliza el nombre del alias indicada en el parámetro –link.

Finalmente por cada variable de entorno definido en el contenedor enlazado, en este caso servidor\_mysql, se crea una en el contenedor principal.



**REFERENCIAS:**

<https://www.josedomingo.org/pledin/2016/02/enlazando-contenedores-docker/>

<https://enmilocalfunciona.io/instalando-y-probando-docker-en-windows-10/>

<https://hub.docker.com/repositories>