**Despliegue de Aplicaciones**

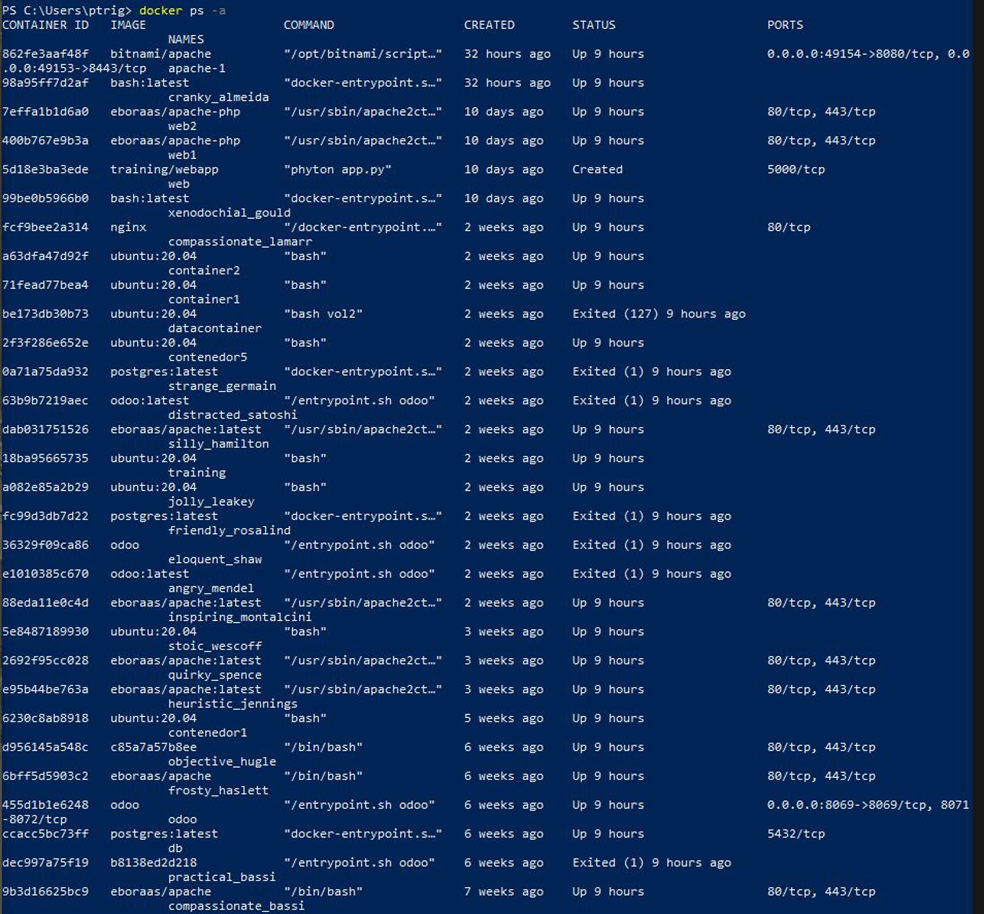
**Servicios de red implicados (Dockers)**

**Tarea Final Trimestre 1**

Imagen que contiene dibujo

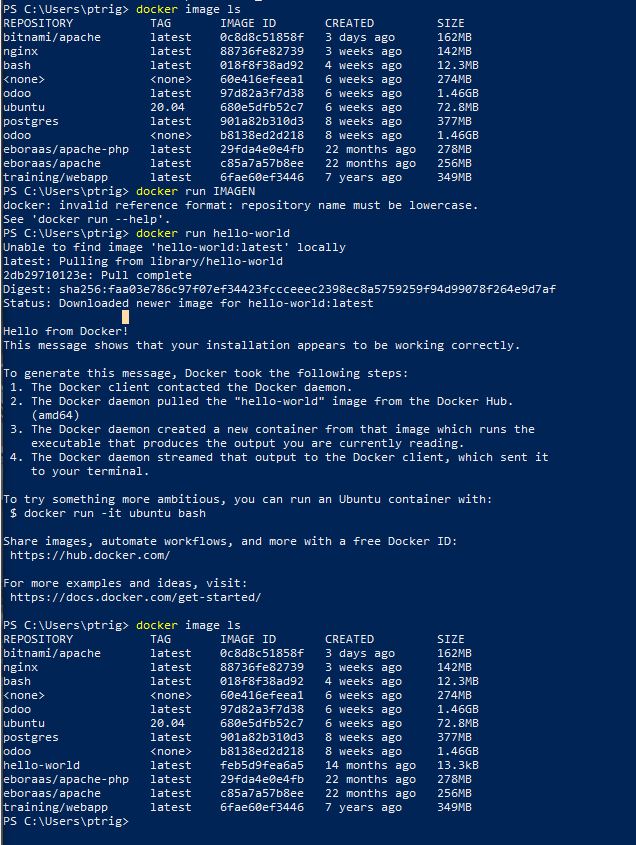
Descripción generada automáticamente

**Tarea Final Trimestre 1 (Parte I): Crea una imagen personalizada a tu gusto**



Comprobar Todos los contenedores que hay en el equipo.

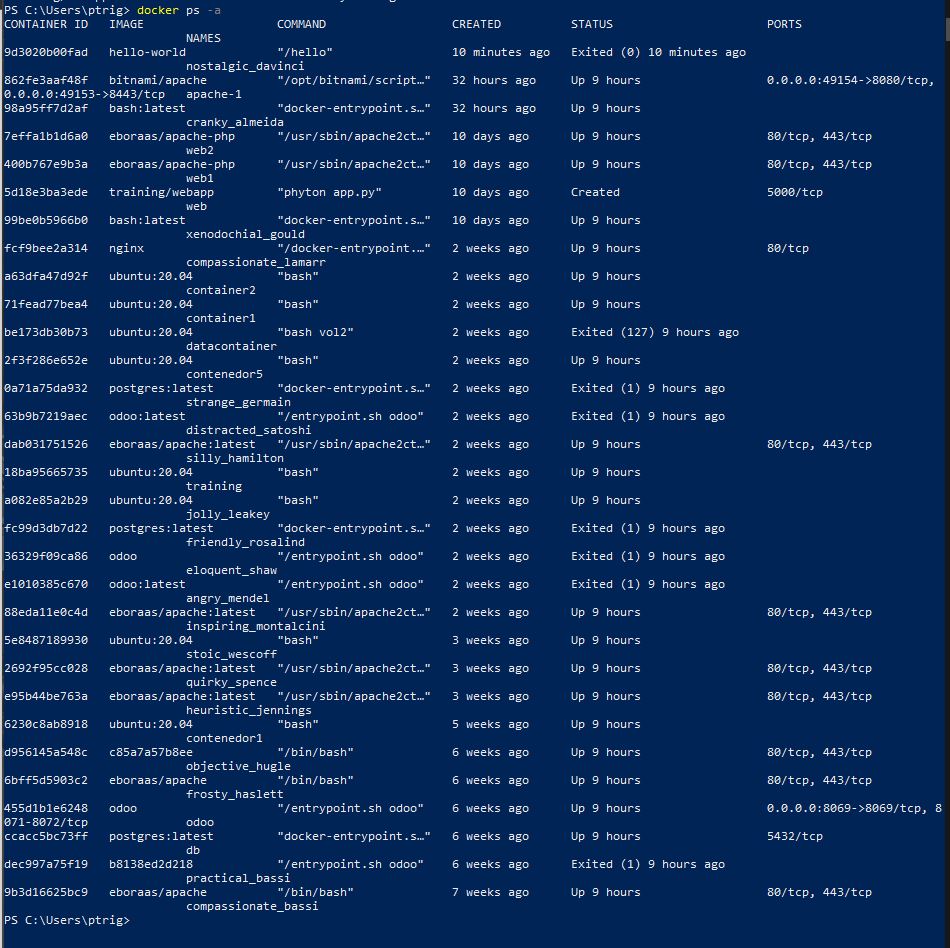
Aun no aparece el contenedor que vamos a crear



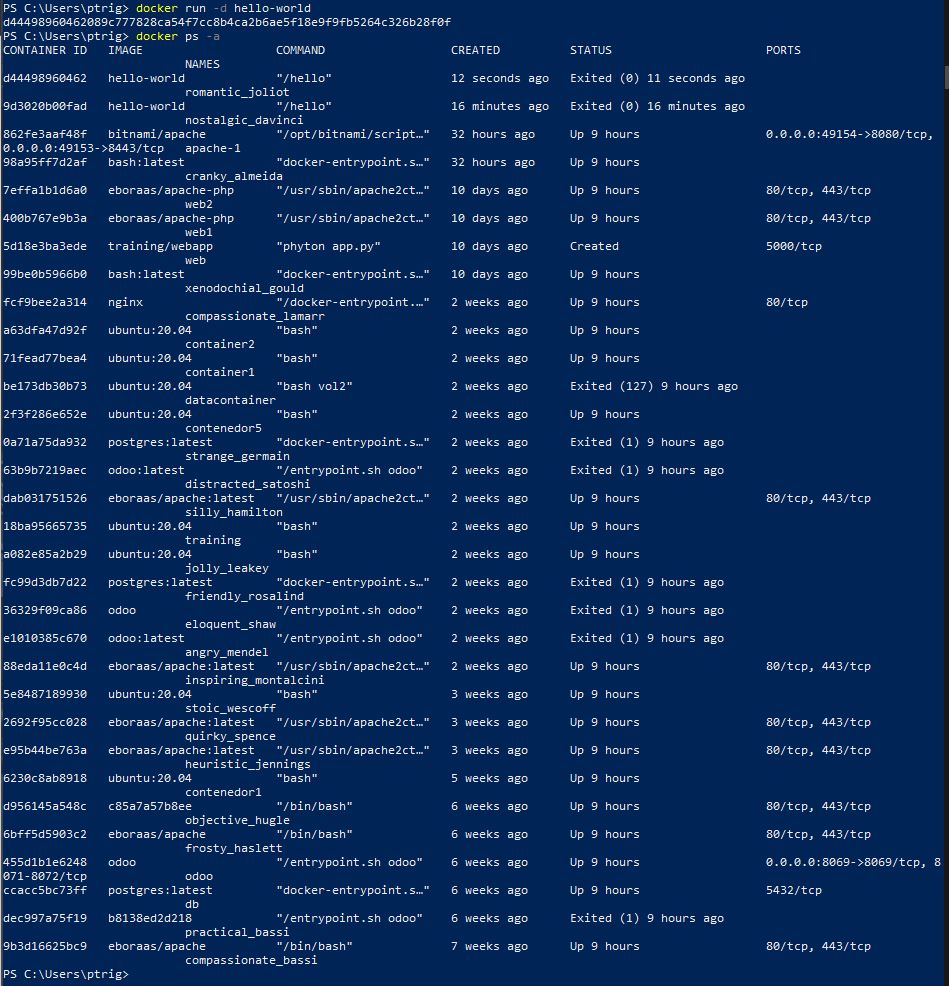
Comprobar las imágenes que tenemos en el equipo.

Crear la imagen hello-world

Comprobar las imágenes que tenemos, ahora con el hello-world incluido.



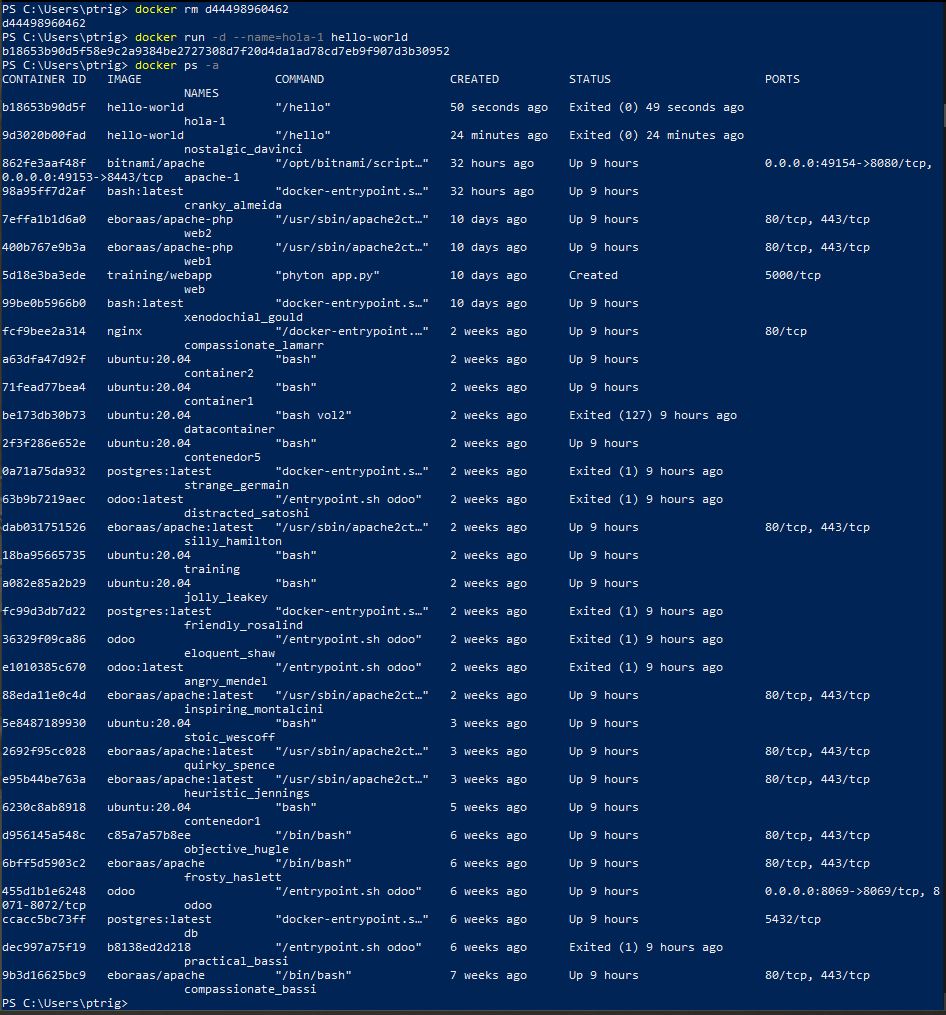
Hacer una comprobación de los contenedores existentes donde esta el hello-world.



Uso la opción **-d**, que arranca el contenedor en segundo plano (*detached*) y permite seguir teniendo acceso a la Shell

Con esta opción solo se muestra el identificador.

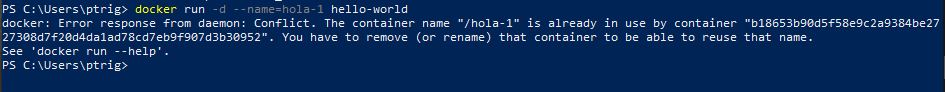
Listamos los contenedores existentes y vemos que hay dos contenedores



Eliminamos unos de los dos contenedores con el comando rm, haciendo referencia a ellos mediante el identificador(id).

Comprobamos que ya no hay contenedores.

El contenedor se muestra con el nombre que hemos indicado (hola-1)



Al intentar crea un contenedor con un nombre ya usado, Docker nos informa de que no es posible hacerlo.



**Tarea Final Trimestre 1 (Parte II)**

**Paso1:**

Crea un contenedor que contenga un servidor Apache. Comprueba si en el navegador se abre la página inicial del contenedor.

Comando: docker run -p 8181:8080 --name paso1 bitnami/apache

****

**Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente**

**Paso 2:**

Genera una nueva imagen, a partir de un Dockerfile, que modifique la página inicial. Crea un contenedor a partir de la nueva imagen y abre en el navegador la página inicial del contenedor para comprobar que se ha modificado.

Creamos el dockerfile y el index.html en la carpeta paso2.

Comando: docker build -t apache\_paso2 .

Comando: docker run -d -p 8282:80 --name paso2 apache\_paso2

**Texto

Descripción generada automáticamente**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

**Paso 3:**

Crea un nuevo contenedor que la página web inicial esté alojada en una ubicación que aunque se elimine el contenedor ésta no se destruya.

Creamos un Dockerfile en el directorio paso3.

-Lo construimos

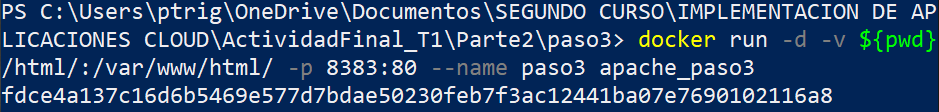
Comando: docker build -t apache\_paso3 .

-Creamos el contenedor

Comando: docker run -d -v ${pwd}/html/:/var/www/html/ -p 8383:80 --name paso3 apache\_paso3

**Texto

Descripción generada automáticamente**

****

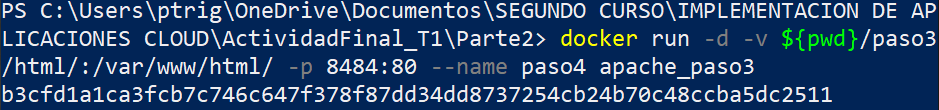
**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

**Paso 4:**

Crea un segundo contenedor que abra la misma página web inicial que el contenedor anterior. Comprueba que la pagina web inicial es la misma que la del contenedor del paso 3.

Comando: docker run -d -v ${pwd}/paso3/html/:/var/www/html/ -p 8484:80 --name paso4 apache\_paso3

****

**Interfaz de usuario gráfica, Texto

Descripción generada automáticamente**

**Paso 5:**  
Los dos últimos contenedores deben poder comunicarse entre sí, sin utilizar la red por

defecto. Realiza la comprobación de que hay comunicación entre ellos.

Hay que modificar el index html de dentro de paso3/html/

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

**Paso 6:**

Elimina el primer contenedor y constata que la página web inicial no se ha eliminado también.

-Para paralo.

Comando: docker stop paso3

-Para eliminarlo.

Comando: docker rm paso3

**Texto

Descripción generada automáticamente**

Comprobar que paso4 sigue teniendo html

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**