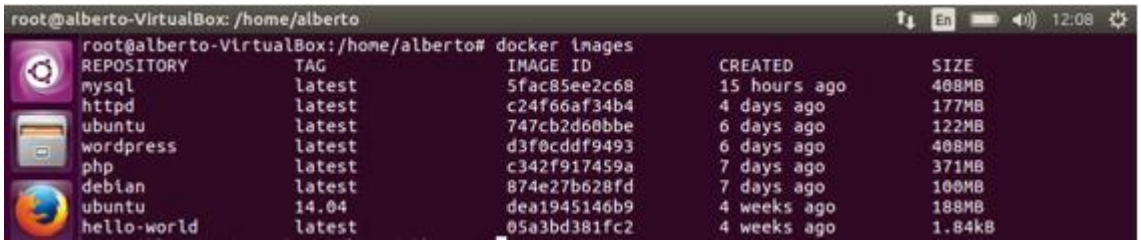
Juan Alberto Peña Núñez
2ºASIR; IAW

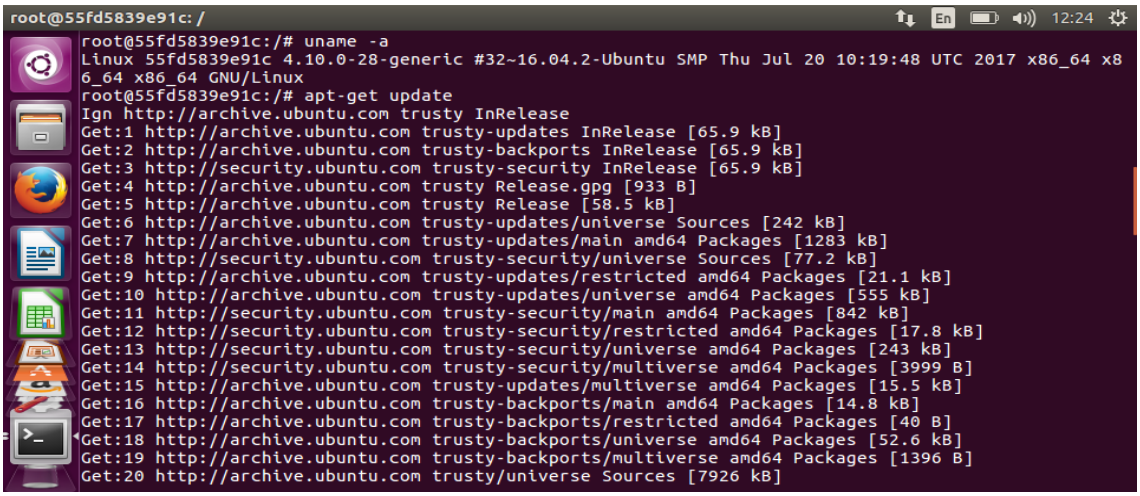
Practicando con Docker

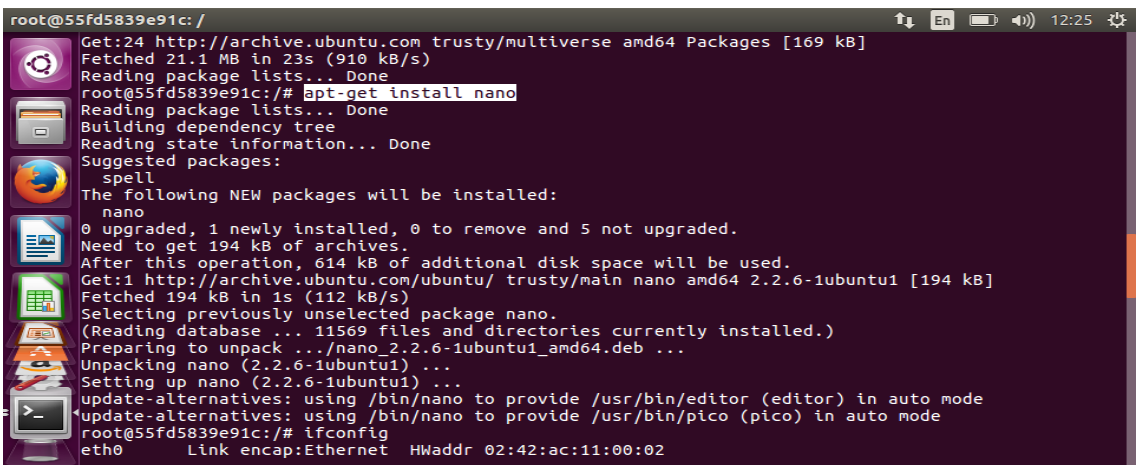
1. Bajar las imágenes que serán la "plantilla" para los contenedores que usaremos posteriormente (usaremos **docker pull**)
2. Obtener la lista de imágenes disponibles



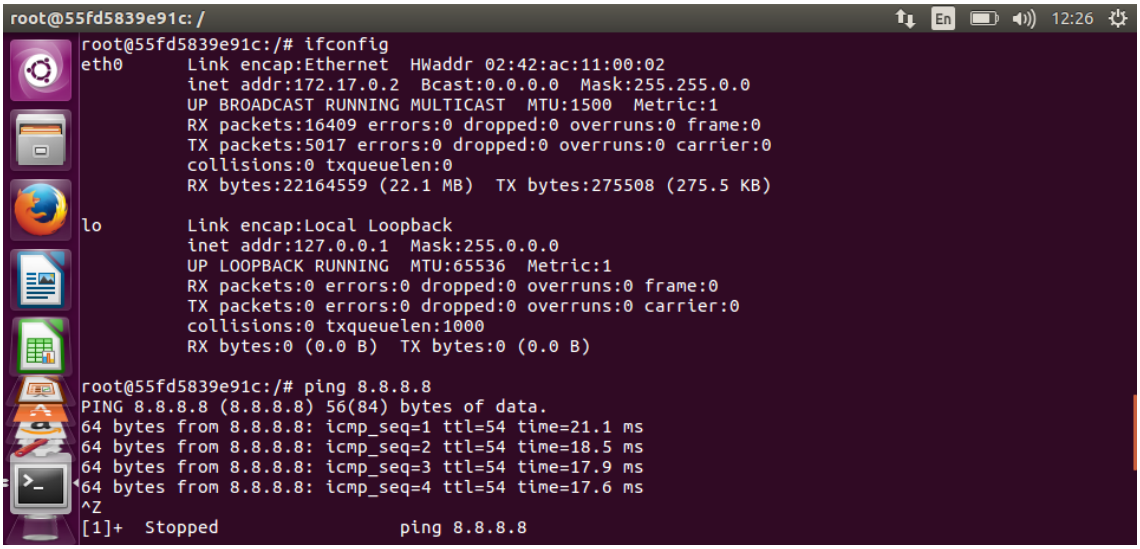
3. Creamos un contenedor partiendo de una imagen (la imagen ubuntu:14.04). Podéis probar con otras imágenes.
   docker run -it --name u1 ubuntu:14.04 /bin/bash

Al arrancar el contenedor obtendremos el prompt de administrador para un sistema Ubuntu 14.04 que vemos arriba
Ejecutar las siguientes ordenes dentro de ese contenedor:

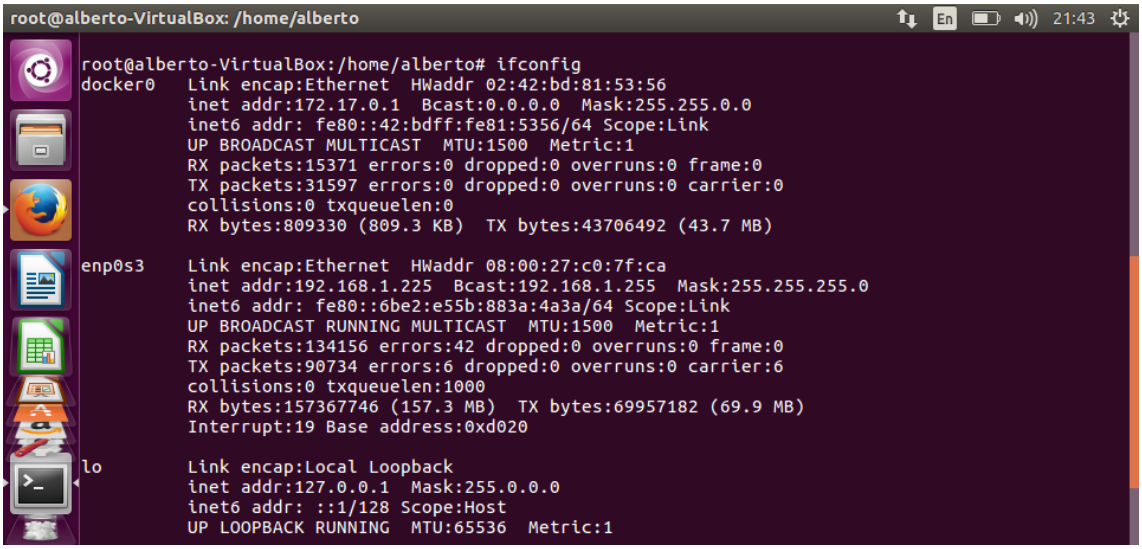




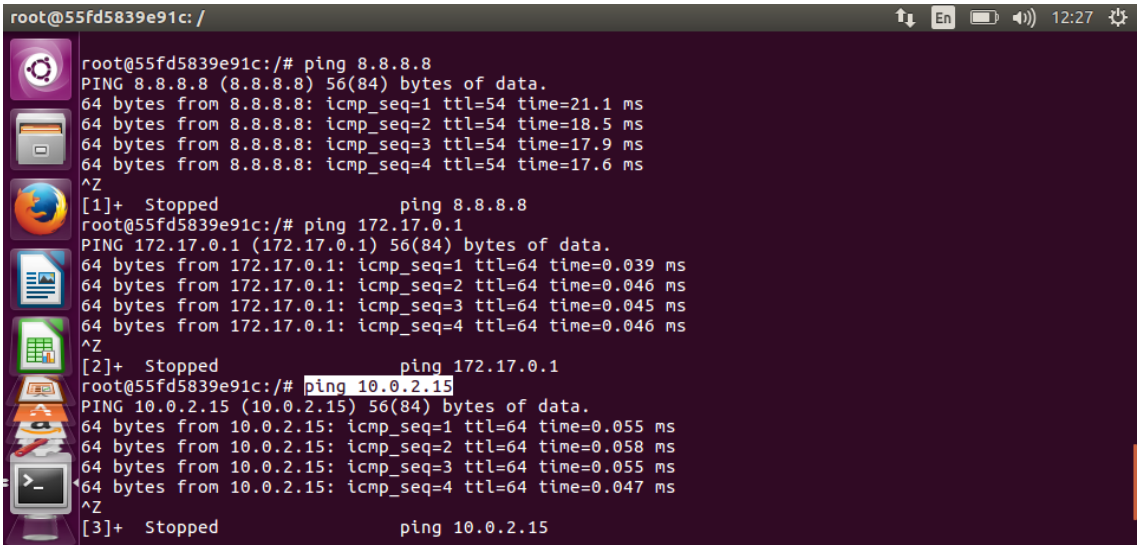
ifconfig (comprobar el tipo de ip que se le ha concedido al contenedor)
ping 8.8.8.8 desde el contenedor (podrás comprobar que funciona)



ifconfig en otra terminal del host docker (fuera del contenedor, en nuestro equipo)



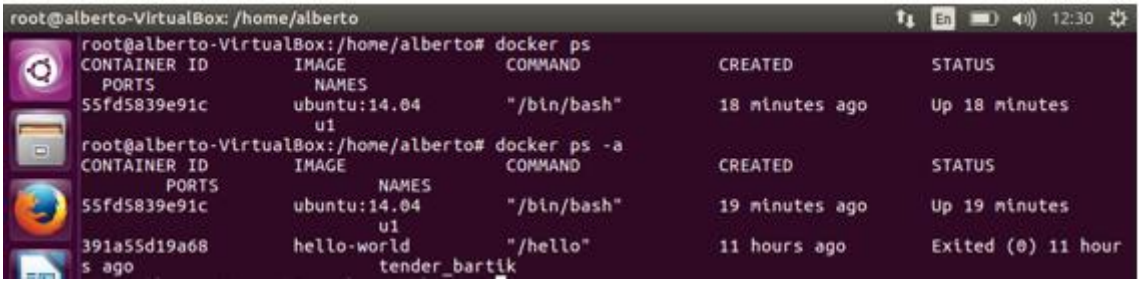
ping a la ip de tu host docker



exit

Hemos salido del contenedor y el contenedor, por cómo está definido y por como lo hemos arrancado dejará de ejecutarse.

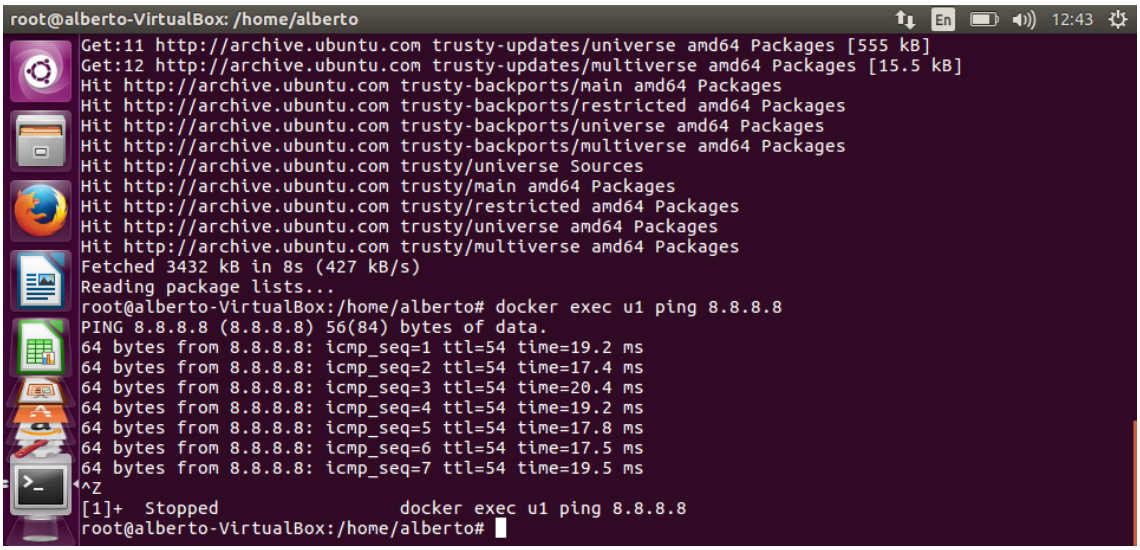
4. Listar los contenedores que están en ejecución



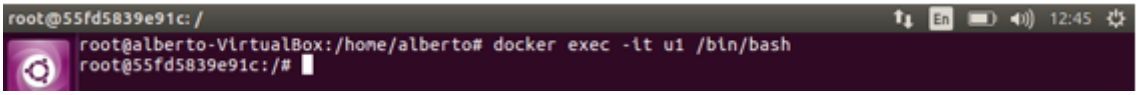
Podemos ver que el estado de nuestro contenedor es Up, es decir, está funcionando.

5. Parar (**docker stop**) y rearrancar (**docker start**) contenedores

6. Ejecutar órdenes en un contenedor que está ejecutándose pero si necesidad de tener que ejecutar un Shell
   Si el contenedor está funcionando (comprueba que está UP con **docker ps**) para ejecutar una orden en ese contenedor usaremos la orden:



Y si queremos lanzar un bash en un contenedor funcionado:



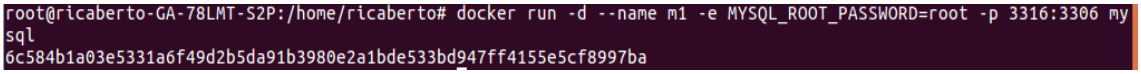
Juan Alberto Peña Núñez
2ºASIR; IAW

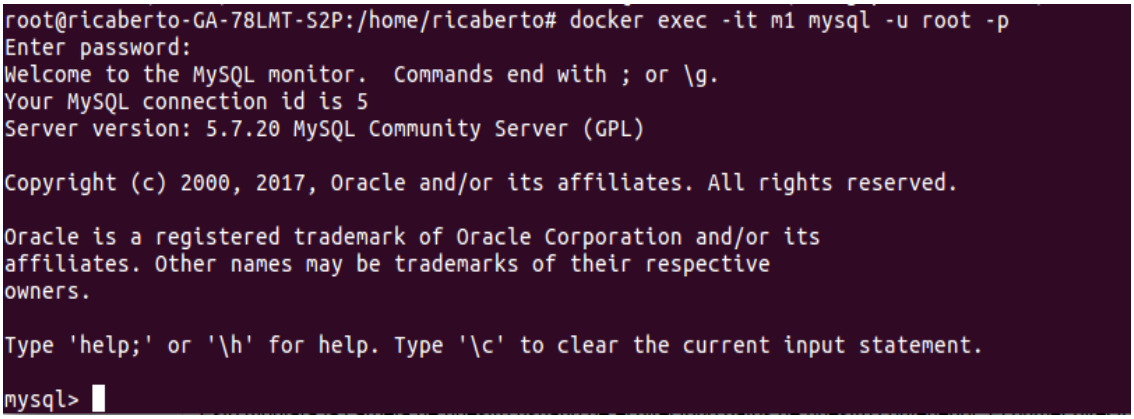
Arrancando contenedor con servicios

1. Arrancar contenedores con servicios

2. Arrancar un contenedor de la imagen de MySQL

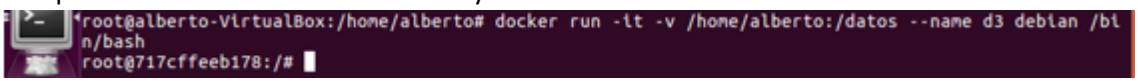
***No se puede realizar en la máquina virtual Ubuntu 16.04 de pruebas debido a problemas con la arquitectura del SO Windows, máquina real sobre la que se ejecuta el software de emulación. Se le adjunta la captura de un compañero en su defecto y a su petición**



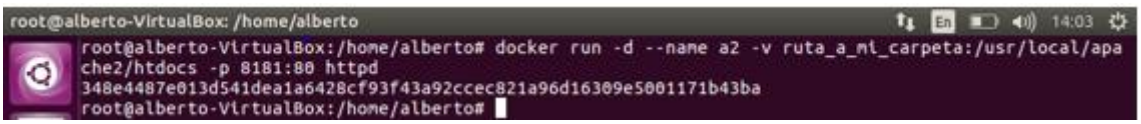


mysql -u root -h X.X.X.X (tu ip) -P 3316 -p

3. Compartiendo directorios entre el host y el contenedor



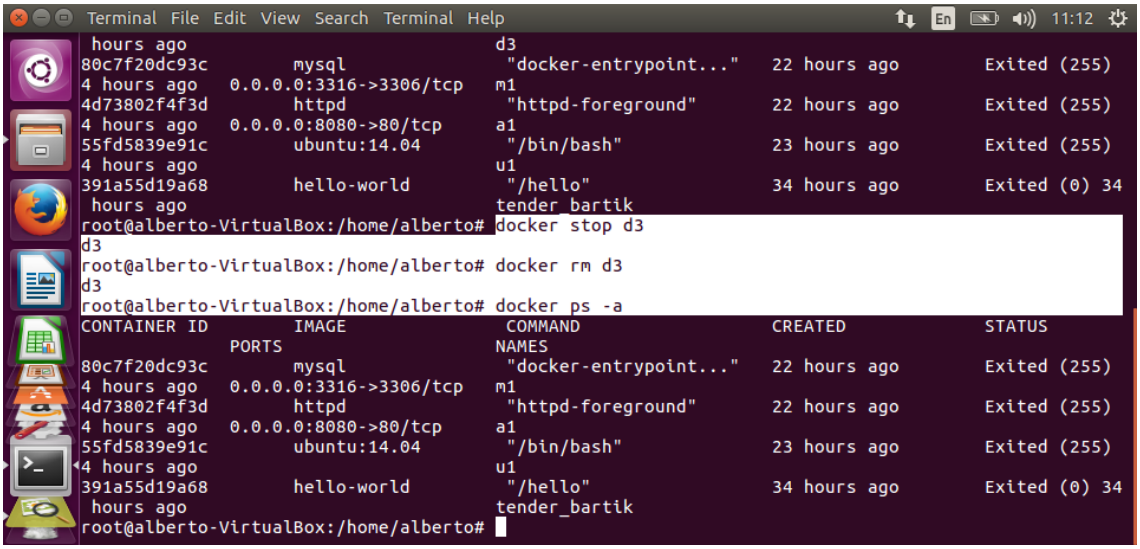
4. Usando una carpeta propia como directorio raíz de un servidor web.



5. Viendo los puertos abiertos y compartidos por cada contenedor

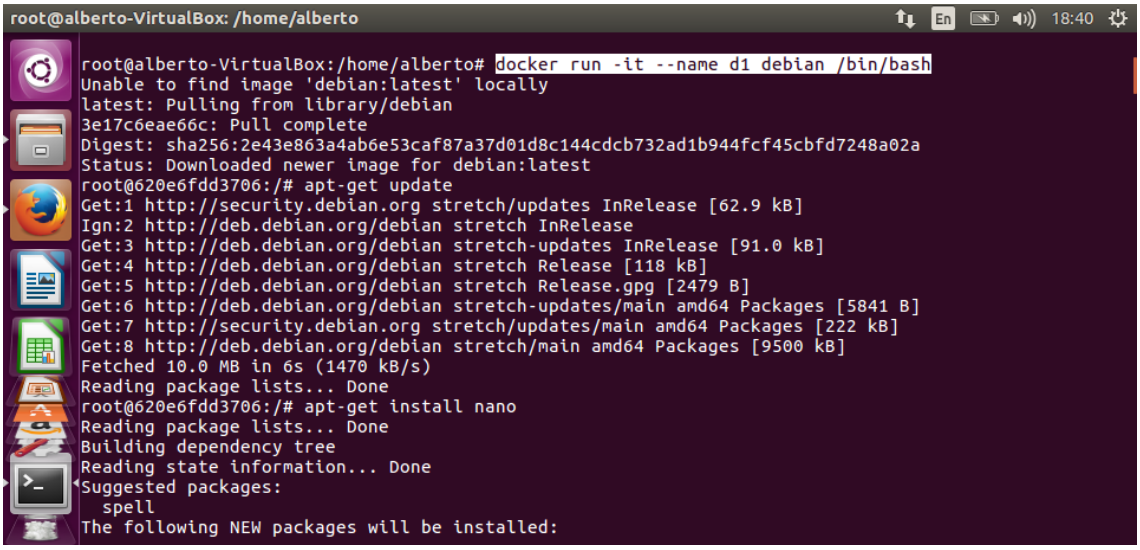
Gestión de imágenes y contenedores

1. Para borrar contenedores
   *Primero se para el contenedor (**docker stop nombre_contenedor**)

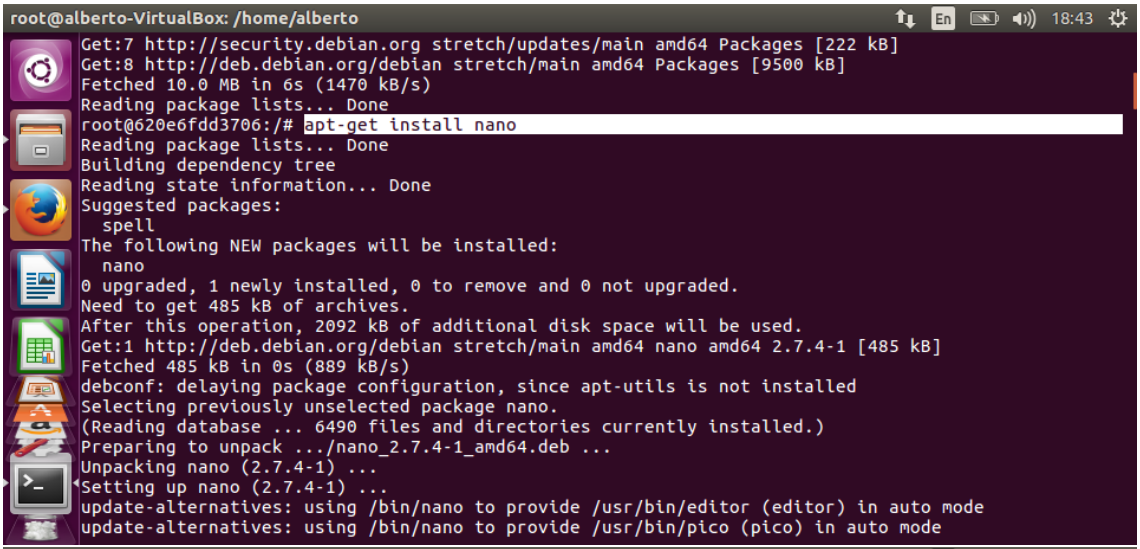


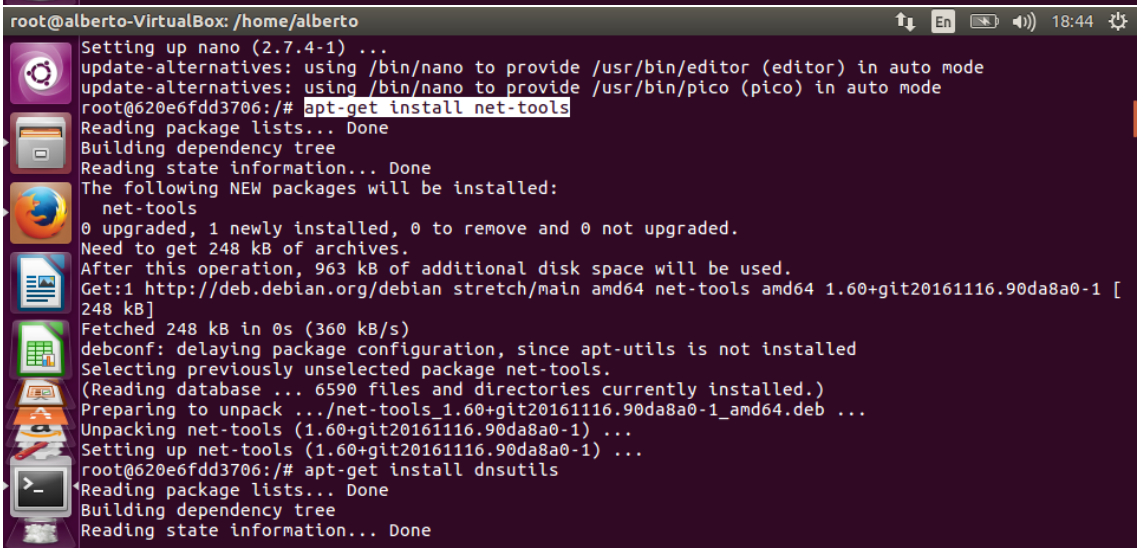
2. Para borrar imágenes

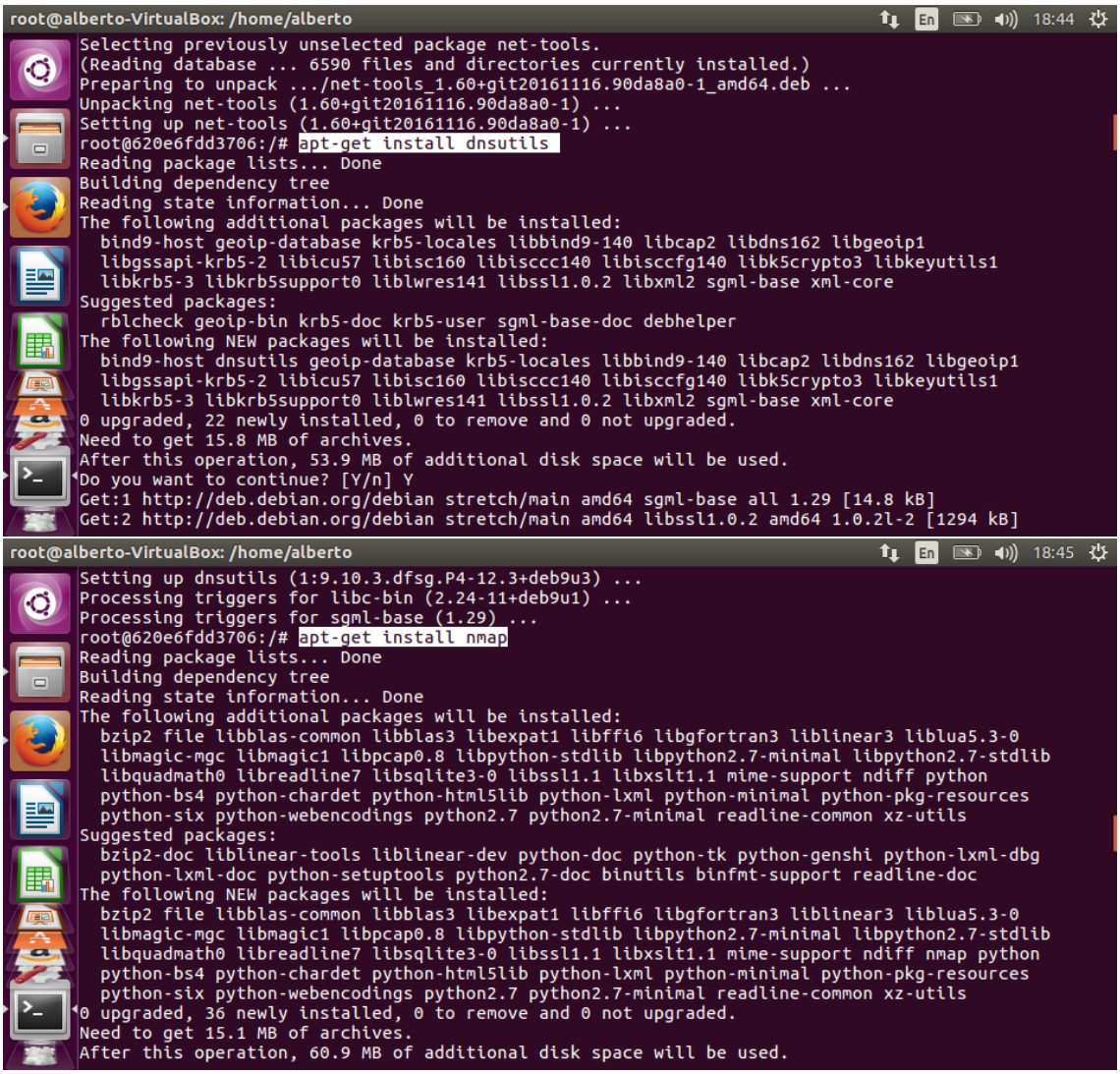
3. Creando una nueva imagen a partir de un contenedor



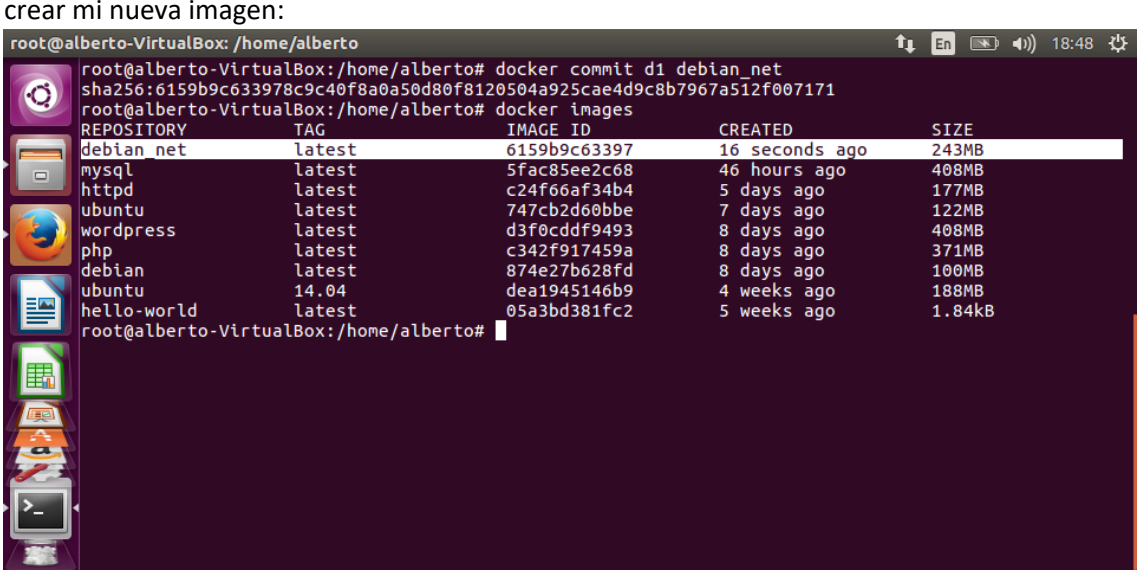
En ese contenedor vamos a ejecutar la siguiente órdenes para instalar ciertos programas para la gestión de red que por defecto no vienen instalador en la imagen de debian:latest





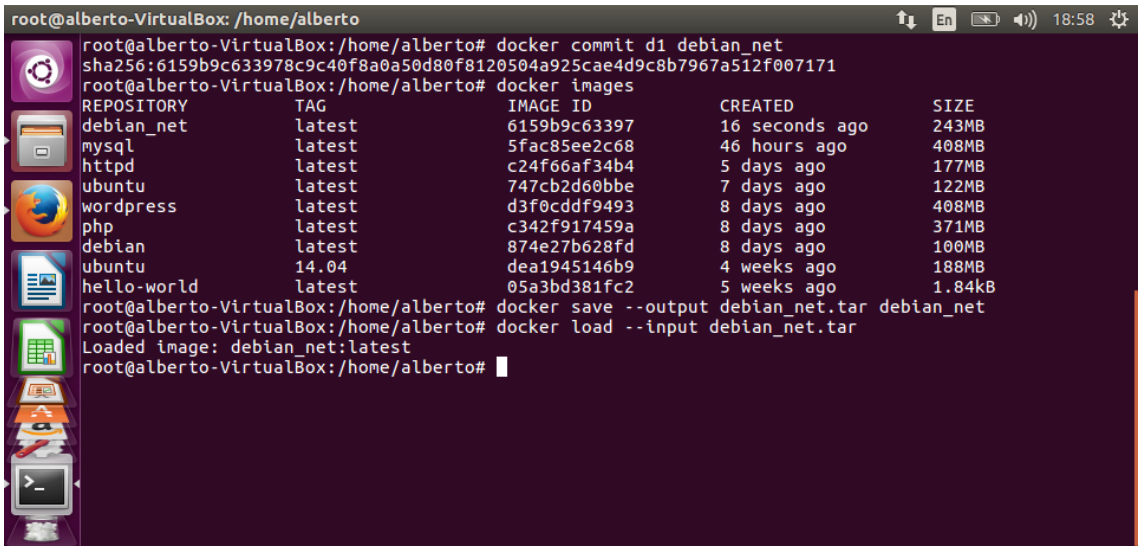


Salgo del contenedor y desde la línea de comandos del host ejecuto esta orden para crear mi nueva imagen:

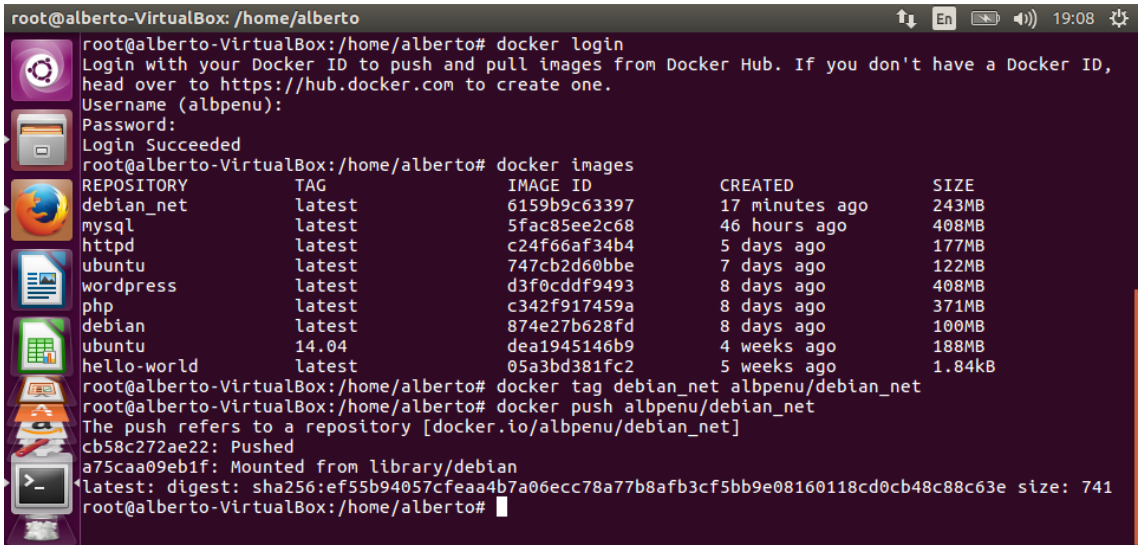


Y al hacer docker images tendré esa nueva imagen en nuestra lista de imágenes.
Para compartirla puedo hacer dos cosas:

a) Guardarla con **docker save** , compartir el fichero y cargarla con **docker load**:



b) Subirla a DockerHub



Y podremos comprobar que esa imagen ya está pública a través de nuestra cuenta de DockerHub.