

Proyecto 02 - Dockerización de un SO

MANUEL CATALÁN, LESTER GARCÍA, EDWIN HILARIO, AXEL RODRÍGUEZ, STEVEN VILLATORO

Universidad Rafael Landívar - Ingeniería en informática y sistemas

Resumen

Un sistema operativo es el programa esencial para cualquier equipo informático. Permite la interacción entre el usuario y el equipo en cuestión por medio de una interfaz gráfica fácil de entender y utilizar. Pero no solo se queda en la interacción con el usuario, también realiza tareas independientes las cuales brindarán una experiencia de uso más optimizada y ordenada de la que el usuario podría imaginar, como gestionar el software, controlar dispositivos externos, almacenar archivos, optimización de uso de memoria y disco, entre otros.

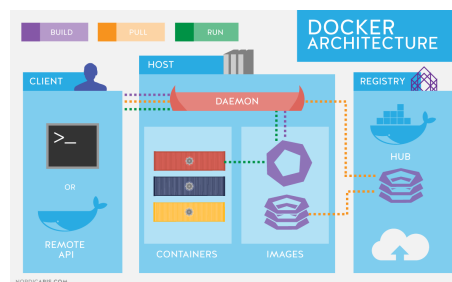
I. INTRODUCCIÓN

EN el área de informática y sistemas, el término virtualización es un método del cual podemos beneficiarnos todos los usuarios sin importar el flujo de trabajo que se esté manejando, ya que en pocas palabras, un sistema de virtualización es un tipo de tecnología que nos permite ejecutar un sistema operativo dentro de otro sistema operativo con todas las ventajas que ello representa y sin tener la necesidad de apagar o reiniciar el equipo para acceder al otro sistema.

Hoy en día, el uso de sistemas virtualizados son un ventana llena de posibilidades en cuanto a trabajo ágil, productividad y eficiencia en las tareas que se buscan desarrollar.

Docker es un proyecto de código libre que se ha convertido en uno de los términos de moda por las ventajas que proporciona, entre otros, a los profesionales del desarrollo web y de aplicaciones, o los administradores de sistemas, por la facilidad que supone el trabajar con el concepto de contenedores. Por otro lado pero sin salir de contexto, DockerHub es un repositorio público en la nube, similar a GitHub, para distribuir información en contenedores.

En la siguiente imagen se puede observar la arquitectura que presenta Docker.



Los casos de uso más comunes hoy en día para Docker y su respectivo servicio de contenedores de imágenes DockerHub son:

- Microservicios
- Aplicaciones híbridas
- Procesamiento de lotes
- Plataformas de servicios múltiples
- Virtualización de imágenes

II. DESCRIPCIÓN DEL PROBLEMA

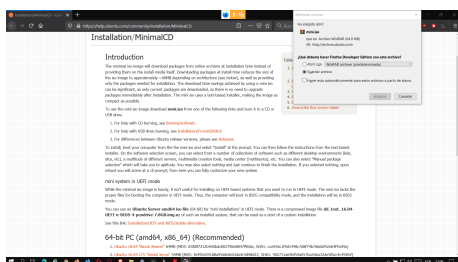
La virtualización de sistemas operativos toma una importancia mayor cuando el manejo de archivos y ciertos programas se vuelvan difíciles o casi imposibles de ejecutar por temas de compatibilidad y recursos del equipo en cuestión para procesarlos.

Para esta ocasión, el problema principal era poder personalizar una imagen de un sistema basado en linux en un contenedor de docker.

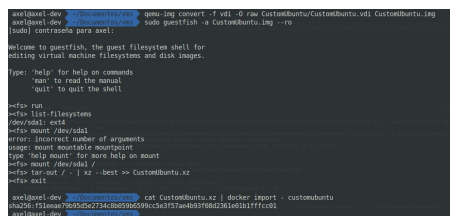
III. TRABAJO REALIZADO

Se definió un pipeline de trabajo necesario para realizar una tarea en particular y se creó una imagen de contenedor el cual fue capaz de ejecutar la tarea. Esta imagen debió de ser compatibilizada por medio de Docker Hub. A continuación, se detallan todos los pasos del trabajo realizado:

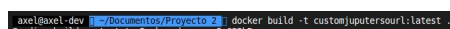
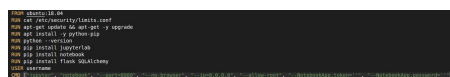
Se descargó una imagen tipo ISO del sistema operativo Ubuntu en su versión 18.04.



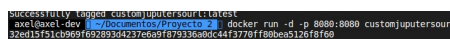
Luego de la creación de la nueva máquina virtual sin interfaz gráfica y de las respectivas modificaciones tanto a nivel de Kernel como a nivel de software instalado se procede a ejecutar los siguientes comandos para convertir la imagen de VirtualBox a una imagen compatible con Docker.



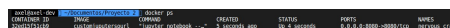
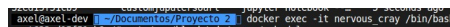
Luego se procede a la construcción de la imagen de docker a partir de un docker file.



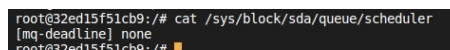
Como siguiente paso, con este comando se ejecuta el contenedor de nombre `customjupyter-sourl` y se mapean puertos, se dice que en el puerto 8080 de la computadora se va a mapear el puerto 8080 de la imagen de docker.



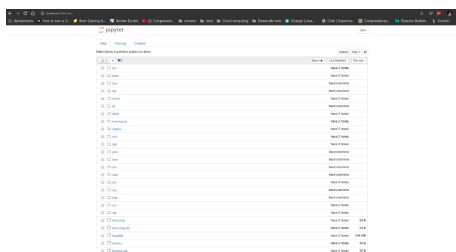
Como siguiente paso, se puede observar el nombre aleatorio que genera Docker para el archivo generado luego de la construcción de la imagen



Por último estando dentro del contenedor se ejecuta el siguiente comando para poder observar el cambio integrado en el Scheduler.



Al finalizar podremos ver la aplicación cargada con un servicio de jupyter notebook previamente instalados en la imagen personalizada del sistema operativo.

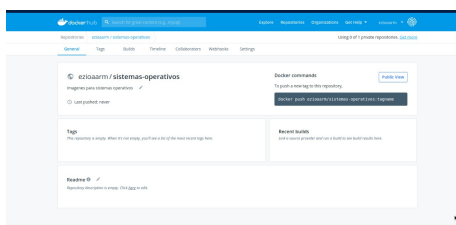


Puntos a tomar en consideración:

Es necesario correr un DockerTag para saber que imagen se estará corriendo en el repositorio creado en DockerHub.

```
axel@axel-dev:~/Documents/vm$ docker tag customjupyter/latest ezioarm/sistemas-operativos:latest
axel@axel-dev:~/Documents/vm$ docker push ezioarm/sistemas-operativos:latest
```

Mi dockerhub, allí dice como hacer push, y lo que va después de docker push es el tag que debe tener la imagen para saber que la tiene que subir.



Antes de lanzar los comandos debemos de realizar un login para evitar errores.

```
axel@axel-dev:~/Documents/vm$ docker login -u "ezioarm" -p "mypassword" docker.io
```

Luego de realizar el login se le realizade nuevo un docker push y se obtendrá el siguiente resultado:

```
axel@axel-dev:~/Documents/vm$ docker push ezioarm/sistemas-operativos:latest
The push refers to repository [docker.io/ezioarm/sistemas-operativos]
90084324fec: Pushed
90084324fec: Pushed
461ee8c478c4: Pushing [=====] 148.7kB
461ee8c478c4: Pushed [=====] 109.9kB/354.7kB
990ef7289899: Pushing [=====]
990ef7289899: Pushed [=====] 156.9kB/1.647GB
```

Por último se realiza un docker pull

ezioarm/sistemas-operativos:latest el cual es para bajar del repo de docker hub.

```
axel@axel-dev:~/Documents/vm$ docker run -d -p 8881:8888 ezioarm/sistemas-operativos
```

IV. CONCLUSIONES

- Un sistema operativo trabaja como la capa intermediaria entre el usuario y todas las funciones que un equipo informático puede llegar a ofrecer
- La correcta utilización de herramientas de virtualización puede optimizar muchas tareas a tal punto que lleguen a poderse trabajar multiplataforma, rompiendo barreras de compatibilidad y brindas seguridad y estabilidad.
- No tener interfaz gráfica de usuario no significa que no sea un trabajo eficiente y funcional, una línea de comandos es igual de eficiente y en cuanto a consumo de recursos estarán mucho más optimizados.

REFERENCIAS

- [1] Sistemas operativos. <https://www.areatecnologia.com/sistemas-operativos.htm>
- [2] VDI a Docker. <https://medium.com/@william.wu/how-to-turn-a-vm-image-into-docker-image-6095f1c63079>
- [3] Virtualización. <https://www.tecnologia-informatica.com/que-es-virtualizacion/>
- [4] VDI to VMDK.. <https://www.dedoimedo.com/computers/virtualbox-convert.html>