Contenedores para incontinentes

Curso 2022-2023

1 Introducción

En este documento vamos a mostrar unos comandos básicos para trabajar con los dockers de la asignatura de SSI. Como ya sabemos la tecnología Docker nos permite simular máquinas virtuales. El proceso consiste en crear imágenes asociadas a cada futura máquina virtual, la ejecución de esas imágenes serán los procesos que simulan cada máquina, lo que denominamos *contenedores*. Cada laboratorio tiene asociados una serie de scripts que nos permitirán llevar a cabo nuestro objetivo. Estos scripts son:

1. **prepare_labX.sh**: encargado de la creación de las imágenes del laboratorio X.
2. **build_labX.sh**: encargado de la ejecución de las imágenes. Primero *arranca* los contenedores y después de un control-C los para y los elimina.
3. **enter_maquina_labX.sh**: nos permite conectarnos a cada máquina en concreto.

Es muy normal que al finalizar una sesión los contenedores queden en ejecución o parados, pero no eliminados completamente. Por este motivo necesitamso conocer unos comandos básicos para trabajar con ellos.

2 Comandos básicos para manejar imágenes

La creación de la imágenes se realiza mediante el script mencionado anteriormente (**prepare_labX.sh**), por lo que realmente no necesitaremos nada más, pero por si acaso, estos comando pueden serte de utilidad para manejar las imágenes:

1. **docker images**: lista la imágenes que tenemos creadas
2. **docker rmi** <**image_id**>: borra la imagen referida
3. **docker run -i -d -t** <**image_id**>: *arranca* el contenedor asociado a esa imagen.

3 Comandos básicos para manejar contenedores

Aunque tenemos un script que arranca y para los contenedores (**build_labX.sh**) y otro que nos permite entrar en las distintas máquinas (**enter_maquina_labX.sh**), no nos viene mal conocer estos comandos para manejar nuestros contenedores *a pelo*:

1. **docker ps**: lista los contenedores en ejecución
2. **docker ps -a**: lista los contenedores en ejecución (Up) y los que están parados (Exited)
3. añadiendo la opción **-q**: se lista únicamente los identificadores de los contenedores, cosa que nos puede ser muy útil para realizar ciertas operaciones sobre un conjunto de contenedores a la vez.
4. **docker stop <container_id>**: para el contenedor referido
5. **docker rm <container_id>**: borra el contenedor referido
6. **docker stop $(docker ps -q)**: para todos los contenedores activos
7. **docker rm $(docker ps -q)**: borra los contenedores parados
8. **docker rm -f $(docker ps -aq)**: para y borra todos los contenedores
9. **docker exec –user ssiuser -it <container_id> /bin/bash**: ejecuta la shell *bash* en el contenedor referido como el usuario *ssiuser*, es decir, *ssiuser* abre una sesión en dicho contenedor.
10. **docker-compose up &**: arranca los contenedores descritos en el fichero *docker-compose.yml* (que debe estar ubicado en el directorio de trabajo)
11. **docker-compose rm** : para y borra los contenedores descritos en el fichero *docker-compose.yml*