

10_docker_container

October 15, 2020

Es posible realizar diversas operaciones de gestión de contenedores mediante el subcomando **docker container**.

Para mayor referencia se puede consultar: <https://docs.docker.com/engine/reference/commandline/container/>

0.1 Contenedores de ejemplo.

Para ilustrar las funciones de gestión de contenedores se utilizarán diversas imágenes.

A continuación se crearán varios contenedores.

Advertencia: Asegúrese de haber creado la imagen *nginx_image* creado al ejecutar la notebook [10_dockerfile.ipynb](#).

- Antes que nada es necesario asegurarse de que no existan contenedores en el sistema, por lo que las siguientes celdas los detendrán y eliminarán.
- En el caso de que no existan contenedores, las celdas enviarán un mensaje de error.

```
[ ]: sudo docker stop $(sudo docker ps -a -q)
```

```
[ ]: sudo docker rm $(sudo docker ps -a -q)
```

```
[ ]: sudo docker ps -a
```

- En caso de que se esté usando esta notebook desde la VM de Cloudevel® es necesario detener el servidor Apache de dicha VM.

```
[ ]: sudo systemctl stop apache2  
sudo systemctl status apache2 --no-pager
```

- Ahora se crearán 6 contenedores, de lo cuales 4 de ellos estarán en ejecución.

```
[ ]: sudo docker run -d -v webroot:/var/www/html -p 80:80 --name servidor nginx_image
```

```
[ ]: sudo docker run -dti --name ubuntu-1 ubuntu /bin/bash  
sudo docker run -dti --name ubuntu-2 ubuntu /bin/bash  
sudo docker run -dti --name ubuntu-3 ubuntu /bin/bash  
sudo docker run --name ubuntu-4 ubuntu echo "Hola"  
sudo docker run --name ubuntu-5 ubuntu echo "Hola otra vez"
```

- Se puede apreciar que existen 4 contenedores ejecutándose.

- *servidor*.
- *ubuntu-1*.
- *ubuntu-2*.
- *ubuntu-3*.

```
[ ]: sudo docker ps
```

- También se puede apreciar que existen en total 6 contenedores en el sistema.
 - *servidor*.
 - *ubuntu-1*.
 - *ubuntu-2*.
 - *ubuntu-3*.
 - *ubuntu-4*.
 - *ubuntu-5*.

```
[ ]: sudo docker ps -a
```

0.2 Comandos informativos.

0.2.1 El comando `docker container ls`.

El comando `docker container ls` regresa un listado de los contenedores de un sistema.

Este comando puede aceptar ciertos argumentos, los cuales pueden ser consultados en https://docs.docker.com/engine/reference/commandline/container_ls/

Ejemplos:

- Al utilizar el comando sin argumentos, se despegará una lista de aquellos contenedores en ejecución.

```
[ ]: sudo docker container ls
```

- Al utilizar el comando con el argumento `-a`, se despegará una lista de todos los contenedores.

```
[ ]: sudo docker container ls -a
```

- Al utilizar la opción `--filter` es posible aplicar un filtro en la búsqueda.

Los filtros disponibles pueden ser consultados en <https://docs.docker.com/engine/reference/commandline/ps/#filters>

```
[ ]: sudo docker container ls --filter name=ubuntu*
```

```
[ ]: sudo docker container ls -a --filter name=ubuntu*
```

0.2.2 El comando `docker container inspect`.

Este comando es idéntico a `docker inspect` y despliega los detalles en formato JSON de uno o más contenedores.

Referencia: https://docs.docker.com/engine/reference/commandline/container_inspect/

Ejemplo:

Se desplegará la información del contenedor con nombre *servidor*.

```
[ ]: sudo docker container inspect servidor
```

0.2.3 El comando `docker container logs`.

Este comando es idéntico a `docker logs` y despliega la bitácora de uno o más contenedores.

Referencia: https://docs.docker.com/engine/reference/commandline/container_logs/

Ejemplo:

Se desplegará la bitácora del contenedor con nombre *servidor*.

```
[ ]: sudo docker container logs servidor
```

0.2.4 El comando `docker container port`.

Este comando es idéntico a `docker port` despliega la configuración de redireccionamiento de puertos de uno o más contenedores.

Referencia: https://docs.docker.com/engine/reference/commandline/container_port/

Ejemplo:

Se desplegará la configuración de reenvío de puertos del contenedor con nombre *servidor*.

```
[ ]: sudo docker container port servidor
```

0.2.5 El comando `docker container top`.

Este comando es idéntico a `docker top` y despliega los procesos ejecutándose al momento en un contenedor.

Referencia: https://docs.docker.com/engine/reference/commandline/container_top/

Ejemplo:

Se desplegará la información de los procesos en ejecución dentro del contenedor con nombre *servidor*.

```
[ ]: sudo docker container top servidor
```

0.2.6 El comando `docker container stats`.

Este comando es idéntico a `docker stats` y despliega las estadísticas de uso de un contenedor de forma continua.

Referencia: https://docs.docker.com/engine/reference/commandline/container_stats/

Ejemplos:

- Al ejecutar el siguiente comando desde una terminal, se desplegarán las estadísticas de todos los contenedores en ejecución.

```
sudo docker container stats
```

- Al ejecutar el siguiente comando desde una terminal, se desplegarán las estadísticas del contenedor *servidor*.

```
sudo docker container stats servidor
```

0.2.7 El comando `docker container diff`.

Este comando es idéntico a `docker diff` y despliega los cambios ocurridos en un contenedor desde su creación en orden cronológico.

Dichos cambios son de 3 tipos a los cuales se les asocia una de las siguientes letras:

- *C* cuando un archivo o directorio es modificado.
- *A* cuando un archivo o directorio ha sido añadido.
- *D* cuando un archivo o directorio ha sido eliminado.

Referencia: https://docs.docker.com/engine/reference/commandline/container_diff/

Ejemplo:

```
[ ]: sudo docker container diff servidor
```

0.3 Comandos de interacción con contenedores.

0.3.1 El comando `docker container attach`.

Este comando es idéntico a `docker attach` y transfiere la entrada y salida estándar de un contenedor a la terminal desde la que se ejecuta el comando.

Referencia: https://docs.docker.com/engine/reference/commandline/container_attach/

Ejemplo:

- Al ejecutar el siguiente comando en una terminal, a la entrada y salida estándar de dicha terminal estará ligada a la entrada y salida estándar del contenedor con nombre *servidor*.

```
sudo docker container attach ubuntu-2
```

0.3.2 El comando `docker container cp`.

Este comando es idéntico a `docker cp` y realiza la copia de archivos entre el sistema de archivos de anfitrión y el de un contenedor.

Referencia: https://docs.docker.com/engine/reference/commandline/container_cp/

Ejemplo:

- El contenedor llamado *servidor* corre un servidor Nginx y despliega el contenido por defecto de dicho servidor.

En caso de estar ejecutando esta notebook desde la VM publicada en el sitio de Cloudevel®, el servicio puede ser accedido desde <http://localhost:8980>.

En caso de estar ejecutando esta notebook desde un sistema local, el servicio puede ser accedido desde <http://localhost>.

- A continuación se realizará le copiará el archivo [src/index.html](#), el cual sustituirá a la página por defecto de Nginx.

ADVERTENCIA: Asegúrese que el servidor Apache de la VM proporcionada por Cloudevel® esté deshabilitado.

```
[ ]: sudo docker container cp src/09/webroot/index.html servidor:/var/www/html/
```

0.3.3 El comando `docker container exec`.

Este comando es idéntico a `docker exec` y envía al contenedor un comando para ser ejecutado por este y la salida estándar de dicho comando será desplegada.

Referencia: https://docs.docker.com/engine/reference/commandline/container_exec/

Ejemplo: La siguiente celda desplegará el listado del directorio `/var/www/html` del contenedor llamado *servidor*.

```
[ ]: sudo docker container exec servidor ls /var/www/html
```

0.4 Comandos de gestión de contenedores.

0.4.1 El comando `docker container rename`.

Este comando es idéntico a `docker rename` y cambia el nombre de un contenedor.

Referencia: https://docs.docker.com/engine/reference/commandline/container_rename/

Ejemplo:

La siguiente celda cambiará el nombre del contenedor *servidor* a *nginx*.

```
[ ]: sudo docker container rename servidor nginx
```

```
[ ]: sudo docker container ls --filter name=nginx
```

0.4.2 El comando `docker container pause`.

Este comando es idéntico a `docker pause` y se encarga de detener todos los procesos en ejecución dentro de un contenedor.

Referencia: https://docs.docker.com/engine/reference/commandline/container_pause/

Ejemplo:

```
[ ]: sudo docker container pause nginx
```

```
[ ]: sudo docker container ls --filter name=nginx
```

0.4.3 El comando `docker container unpause`.

Este comando es idéntico a `docker unpause` y reinicia la ejecución de un contenedor pausado.

https://docs.docker.com/engine/reference/commandline/container_unpause/

Ejemplo:

```
[ ]: sudo docker container unpause nginx
```

```
[ ]: sudo docker container ls --filter name=nginx
```

0.4.4 El comando `docker container stop`.

Este comando es idéntico a `docker stop` y envía una señal de apagado de un contenedor en ejecución.

https://docs.docker.com/engine/reference/commandline/container_stop/

Ejemplo:

```
[ ]: sudo docker container stop ubuntu-1
```

```
[ ]: sudo docker container ls --filter name=ubuntu-1
```

0.4.5 El comando `docker container start`.

Este comando es idéntico a `docker start` e inicia la ejecución de un contenedor.

https://docs.docker.com/engine/reference/commandline/container_start/

Ejemplo:

```
[ ]: sudo docker container start ubuntu-1
```

```
[ ]: sudo docker container ls --filter name=ubuntu-1
```

0.4.6 El comando `docker container restart`.

Este comando es idéntico a `docker restart` y reinicia a un contenedor en ejecución.

https://docs.docker.com/engine/reference/commandline/container_restart/

Ejemplo:

```
[ ]: sudo docker container restart ubuntu-1
```

0.4.7 El comando `docker container kill`.

Este comando es idéntico a `docker kill` y destruye a un contenedor. Se utiliza cuando un contenedor no puede ser detenido mediante el comando `docker container stop`.

https://docs.docker.com/engine/reference/commandline/container_kill/

Ejemplo:

```
[ ]: sudo docker container kill ubuntu-1
```

0.4.8 El comando `docker container rm`.

Este comando es idéntico a `docker rm` y elimina a un contenedor.

https://docs.docker.com/engine/reference/commandline/container_kill/

Ejemplo:

```
[ ]: sudo docker container rm ubuntu-1
```

```
[ ]: sudo docker container ls -a --filter name=ubuntu*
```

0.4.9 El comando `docker container prune`

https://docs.docker.com/engine/reference/commandline/container_prune/

```
[ ]: sudo docker ps -a
```

```
[ ]: sudo docker container prune -f
```

```
[ ]: sudo docker ps -a
```

0.4.10 El comando `docker container export`.

Este comando es idéntico a `docker export` y su función es crear un archivo que contengan la información necesarioa para reconstruir un contenedor.

https://docs.docker.com/engine/reference/commandline/container_export/

```
[ ]: sudo docker container export nginx -o nginx.tar
```

```
[ ]: ls *.tar
```

0.4.11 El comando `docker container commit`.

Este comando es idéntico a `docker commit`.

https://docs.docker.com/engine/reference/commandline/container_commit/

```
[ ]: sudo docker commit nginx nginx:123
```

```
[ ]: sudo docker images
```

Esta obra está bajo una Licencia Creative Commons Atribución 4.0 Internacional.

© José Luis Chiquete Valdivieso. 2020.