Principales acciones con Docker

En este apartado vamos a empezar a realizar acciones con Docker. Para ello seguiremos este documento.

Principales acciones con Docker

Para ir recordando los distintos comandos que vayamos usando imprime y ve marcando los comandos que uses en este CheatSheet:

Cheatsheet

Docker exec

En el apartado 9. E JECUTANDO COMANDOS EN UN CONTENEDOR CON “ DOCKER EXEC ” nos explica cómo hacerlo pero no nos guía para probarlo. Vamos a verlo aquí.

Vamos primero a crear un contenedor ubuntu y a lanzar una terminal.

```
docker run -it --name=ubuntu_pruebas ubuntu /bin/bash
root@e0f7b22e64d7:/#
```

Fíjate que el prompt ahora es "root@e0f7b22e64d7:/#", es decir, estamos trabajando dentro de nuestro contenedor. Vamos a ver qué hay dentro del directorio /tmp

```
root@e0f7b22e64d7:/# ls /tmp
```

Está vacío.

Ahora abre en tu máquina host otra terminal y conéctate por ssh a la máquina virtual AWS en la que estamos trabajando. Desde este otro terminal vamos a lanzar comandos a nuestro contenedor `ubuntu_pruebas`. A este otro terminal le llamaremos `terminal2` y al primero, en el que estamos trabajando directamente dentro del contenedor `terminal1`. Vamos a ejecutar en `terminal2` el primer comando que nos indica nuestro manual:

```
docker exec -d ubuntu_pruebas touch /tmp/prueba
```

En este ejemplo se ejecuta ejecuta en “background”, gracias al parámetro “-d”. Este ejemplo simplemente crea mediante el comando “touch” un fichero “prueba” en “/tmp”.

Ahora vuelve a `terminal1` y vuelve a comprobar si hay algo en /tmp

```
root@e0f7b22e64d7:/# ls /tmp
prueba
```

Ahora deberías tener el fichero prueba que creaste con `docker exec`.

Siguiendo con los comandos del manual vamos a conectarnos al contenedor en `terminal2`. Luego salimos con `exit`.

```
docker exec -it ubuntu_pruebas bash
root@e0f7b22e64d7:/# exit
exit
```

Esta orden que ejecutará la “shell” bash en nuestra consola (gracias al parámetro “-it” se enlaza la entrada y salida estándar a nuestra terminal). A efectos prácticos, con esta orden accederemos a una “shell” bash dentro del contenedor.

Seguimos con las pruebas. Vamos a crear una variable de entorno en el contenedor. Comando que establece un variable de entorno con el parámetro “-e” . Se enlaza la entrada y salida de la ejecución del comando con “-it” . A efectos prácticos, en esa “shell” estará disponible la variable de entorno “VAR” con valor 1. Lo podemos probar con “echo $VAR”.

```
$ docker exec -it -e VAR=1 ubuntu_pruebas bash
root@e0f7b22e64d7:/# echo $VAR
1
root@e0f7b22e64d7:/# exit
exit
$
```

Docker cp

En el apartado 10. COPIANDO FICHEROS ENTRE ANFITRIÓN Y CONTENEDORES CON “ DOCKER CP ” nos explica como copiar ficheros entre la máquina anfitrión y el contenedor. Sustituye en los comandos `idcontainer` por el nombre que le hemos dado a nuestro contenedor `ubuntu_pruebas` para probar los distintos comandos. Puedes lanzar los comandos en `terminal2` y probar que los ficheros se guardan en el contenedor en `terminal1`.

Docker attach

Para probar los comandos del apartado 11. A CCEDIENDO A UN PROCESO EN EJECUCIÓN CON “ DOCKER ATTACH ” sal del contenedor `ubuntu_pruebas` en `terminal1`. Ahora no deberías estar dentro de ningún contenedor en ninguno de los 2 terminales que tienes abiertos.

En `terminal1` lanzaremos el comando que nos propone el manual:

```
docker run -d --name=muchotexto busybox sh -c "while true; do $(echo date) ;
sleep 1; done"
```

Ahora en `terminal2` ejecuta el comando:

```
docker attach muchotexto
```

Vemos cómo hemos anlazado la salida del contenedor `muchotexto` a la salida estandar del `terminal2` y vemos la salida del comando que se está ejecutando en el contendor.

Docker logs

Para probar este comando mantén el contenedor `muchotexto funcionando`. En `terminal2` ejecuta:

```
docker logs -f --until=2s muchotexto
```

Una vez finalices la parte teórica prueba a realizar esta práctica para afianzar conocimientos:

Caso práctico 01- Práctica de comandos en contenedor Docker