



UT 11. DOCKER Actividades

Sistemas informáticos CFGS DAW

Álvaro

Maceda

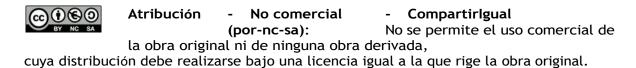
a.macedaarranz@edu.gva.es

2022/2023

Versión:230309.1334

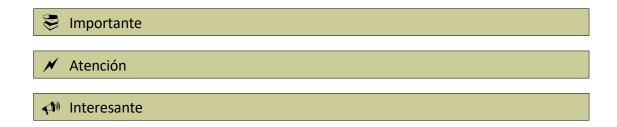
SISTEMAS UD011. DOCKER

Licencia



Nomenclatura

A lo largo de esta unidad se utilizarán diferentes símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:



SISTEMAS UD011. DOCKER

UT 11. ACTIVIDADES

DOCKER

Necesitarás tener instalado Docker en tu máquina para poder realizar estos ejercicios. Si no quieres instalar Docker en tu ordenador, recuerda que puedes utilizar una máquina virtual e instalar Docker allí.

Echa un vistazo a menudo a tus contenedores con docker ps -a hasta que puedas entender fácilmente lo que está pasando.

4. EJERCICIO 4

Busque en Docker Hub y descargue una imagen oficial de Docker para Java JDK, versión 11, proporcionada por IBM (ibmjava)

Crea un contenedor y comprueba que el comando javac existe en el contenedor. No olvides eliminar el contenedor después de eso.

Elimina la imagen de tu ordenador.

5. EJERCICIO 5

Crea una imagen para contenedores que ejecute el comando sla al iniciarse. Necesitarás instalar el paquete sla para tener ese comando disponible. Instalará el comando en el directorio /usr/games.

6. EJERCICIO 6

Parte 1

Crear una imagen basada en un contenedor python: 3 con un archivo llamado reverse.py en el directorio

/usr/bin. El archivo tendrá este contenido:

```
importar sistema

param = sys.argv[1]
print(param[::-1])
```

Parte 2

Modifica la imagen para que el script se lance con la cadena ¡Yo, banana boy! cuando se ejecute un contenedor que utilice esa imagen (debería imprimir ! yob ananab ,oy):

```
docker run --rm <nombre de la imagen>
!yob ananab ,oY
```

SISTEMAS UD011. DOCKER

Parte 3

Modifique la imagen para que el script se ejecute utilizando un parámetro proporcionado en la línea de comandos. Por ejemplo, cuando se ejecuta con:

```
docker run --rm <nombre de la imagen> Kayak
```

Debería imprimir kayaK.

7. EJERCICIO 7

Parte 1

Cree una imagen etiquetada ubuntu-net, basada en ubuntu:latest, que tenga instalados los siguientes paquetes de herramientas de red: iproute2 (para el comando ip), iputils-ping (para ping) y net-tools (para ifconfig).

Parte 2

Lanza cuatro contenedores: contenedor_a, contenedor_b, contenedor_c y contenedor_d, basados en la imagen anterior:

- Los contenedores container_a y container_b deben poder comunicarse entre ellos a través de la red, pero no con container c y container d
- Los contenedores container_c y container_d deben poder comunicarse entre ellos pero no con container a y container b .

Prueba las conexiones usando ping con las direcciones de los contenedores (ping desde el contenedor_a al contenedor_b, y desde el contenedor_a al contenedor_c) Por ejemplo, si el contenedor contenedor b tiene la IP 5.6.7.8 ejecuta ping 5.6.7.8 desde el contenedor a.

Pruebe las conexiones utilizando ping con los nombres de los contenedores. Por ejemplo, ejecute ping B desde

contenedor a.