



UT 11.

DOCKER

Actividades

Sistemas informáticos
CFGS DAW

Álvaro

Maceda

a.macedaarranz@edu.gva.es

2022/2023

Versión:230227.0717

Licencia



Atribución - No comercial - Compartir igual
(por-nc-sa): No se permite el uso comercial de la obra original ni de ninguna obra derivada, cuya distribución debe realizarse bajo una licencia igual a la que rige la obra original.

Nomenclatura

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



Importante



Atención



Interesante

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.

0. EJERCICIO 0

Ejecute un contenedor de la imagen `hello-world` desde su ordenador. Esto le ayudará a probar su instalación Docker.

Recuerde ejecutar `docker ps -a` después de eso.

1. EJERCICIO 1

Haz lo siguiente utilizando los comandos docker `create` y `docker start`. También necesitarás lanzar comandos adicionales:

1. Crea tres ficheros llamados `index.html` en tres directorios diferentes. El contenido de los ficheros será así (cambie `1` y `PRIMERO` por el ordinal correspondiente):

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Página 1</title>
</head>
<body>
  <h1>Soy el PRIMERO servidor web</h1>
</body>
</html>
```

`mkdir <nombre proyecto>`
`touch index.html`
`nano index.html`

 escribimos el código

`"ctrl+x"`
`guardar "Y"`
`confirmar nombre "intro"`

2. Inicie tres servidores web HTTP diferentes utilizando la imagen oficial

httpd: https://hub.docker.com/_/httpd

3. Abra las tres páginas web al mismo tiempo en su navegador

4. Detener y destruir los contenedores

`sudo docker stop <nombre proyecto>`
`sudo docker rm <nombre proyecto>`

Repite el ejercicio, pero esta vez utilizando `docker run` en lugar de `docker create` + `docker start`.

Se utilizará el siguiente código sustituyendo al `create` y al `start`
`sudo docker run -dit --name <nombre proyecto> -p <localhost:puerto> <nombre proyecto>`

2. EJERCICIO 2

Necesita resolver un error en un antiguo script de Python, escrito en la versión 2.7.18 de Python. La versión corregida del script es esta:

```
importar sistema
```

```
sudo docker create --name python2718 -ti python:2.7.8
sudo docker start python2718
sudo docker exec -it python2718/bin/bash
```

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

escribimos el código
ponemos python reverse.py "Esto es una cadena"
y nos devolverá anedac anu se otse

El script debe lanzarse con `python reverse.py "Esto es una cadena"`.

Abre un shell en un contenedor `python:2.7.18`, escribe el script y prueba que funciona (necesitarás instalar un editor de texto como `vim` o `nano`) Intenta abrir un shell para editar el archivo y otro para ejecutar el script.

Destruye el contenedor y crea uno nuevo para probarlo de nuevo. ¿Está ahí tu script? ¿Necesitas reinstalar el editor de texto? `no porque has borrado el contenedor`

3. EJERCICIO 3

Parte 1

En este ejercicio, ejecutarás un servidor de base de datos y crearás una base de datos utilizando Docker.

1. En primer lugar, cree y ejecute un contenedor de servidor MySQL con

la imagen oficial: https://hub.docker.com/_/mysql

NO lo ejecute en modo independiente: un nuevo servidor MySQL tarda un rato en inicializarse la primera vez que se ejecuta y tendrá que esperar hasta que la inicialización se complete (con un mensaje como "listo para conexiones").

Una vez que el servidor esté funcionando, tendrá que probarlo. Para ello, necesitará un cliente MySQL.

2. El servidor ya tiene un cliente instalado. Abra un intérprete de comandos en el servidor y conéctese al servidor de bases de datos ejecutando este comando en el intérprete de comandos del servidor:

```
mysql --user root --password=secret
```

Una vez conectado, puede mostrar las bases de datos y crear una con estos comandos:

```
mostrar bases de datos;  
crear base de datos  
mi_base_de_datos; mostrar bases
```

Puedes crear algunas tablas y rellenarlas si quieres.

3. Ahora, prueba el servidor desde tu ordenador. Instala `mysql-client` en tu sistema. Para conectarte a un servidor MySQL, puedes usar este comando:

```
mysql --host=127.0.0.1 --port 3306 --user=root --password=secret
```

Ese comando se conectará a un servidor de base de datos en tu ordenador, escuchando en el puerto 3306. Tendrás que redirigir ese puerto desde el servidor a tu ordenador, o el cliente no podrá llegar al servidor.

Crear otra base de datos

4. Detenga y destruya el servidor. Vuelve a conectarte al servidor. ¿Están sus bases de datos?

Parte 2

Quieres conservar tu base de datos con todos tus datos cuando se destruya el contenedor:

1. Cree un volumen Docker para su servidor de base de datos
2. Inicie el servidor de base de datos montando el volumen en `/var/lib/mysql`
3. Cree una base de datos en el servidor, utilizando el método que prefiera
4. Destruye el contenedor y lanza uno nuevo montando el volumen en el mismo directorio:
 - ¿Se ha inicializado más rápido el nuevo contenedor?
 - ¿Está ahí tu base de datos?

Puedes comprobar el espacio que utiliza cada imagen, contenedor y volumen en tu sistema con `docker system df -v`. Ejecútalo y mira lo que tienes ahí.

Recuerda destruir el volumen una vez hayas terminado el ejercicio. Ejecuta `docker volume ls` para comprobarlo.

Parte 3

Esta vez, utilizarás montajes bind para almacenar la base de datos:

1. Crea un directorio vacío en tu máquina
2. Inicie el servidor de base de datos montando el directorio de su host en `/var/lib/mysql`
3. Cree una base de datos en el servidor, utilizando el método que prefiera
4. Destruye el contenedor y lanza uno nuevo montando el volumen en el mismo directorio.
¿Está ahí tu base de datos?
5. ¿Quién es el propietario de los archivos creados en su ordenador? ¿Por qué?