

Tarea Evaluable Docker - 2EV

Módulo Despliegue de Aplicaciones Web - CIFP La Laboral

Tarea Evaluable Docker - 2EV

Instrucciones

Ejercicio 1 - Contenedores en red y Docker Desktop

Ejercicio 2 - Docker Compose

Ejercicio 3 - imagen con Dockerfile - Aplicación web

Videoclip

Calificación

Instrucciones

- Se debe crear un repositorio GitHub - *debe ser público para que yo tenga acceso*
 - Se debe trabajar con ramas, una por ejercicio, que se irán fusionando con la rama `main` o principal
 - El contenido del repositorio estará dividido en carpetas, una para cada ejercicio de la tarea
 - En cada carpeta estará el documento/s `.md` que resuelve el ejercicio, las imágenes necesarias y el `.pdf`. Añadir también los ficheros de código utilizados, si procede.
- Se entregará un documento `.pdf` con el nombre de la persona que realiza la tarea, la URL del repositorio donde está la resolución de los ejercicios y la URL del videoclip
 - Opcionalmente, se puede entregar también un fichero comprimido con todos los ficheros `.pdf` generados
- Se valorará la claridad en la organización del repositorio y la completitud de la documentación de las soluciones a los ejercicios

Ejercicio 1 - Contenedores en red y Docker Desktop

Este ejercicio se resolverá con Docker Desktop, **si se soluciona con comandos la nota será 0**

Si necesitas usar comandos, usa la Terminal integrada en Desktop

SUGERENCIA: Utiliza la extensión **PortNavigator** para gestionar las redes en Docker Desktop

1. Crea una red bridge **redej1**
2. Crea un contenedor con una imagen de **mariaDB** que estará en la red **redej1**. Este contenedor se ejecutará en segundo plano, y será accesible a través del puerto 3306.
 - Definir una contraseña para el usuario **root**, y un **usuario con tu nombre de pila** y con contraseña. La BD por defecto será **DAW**
 - Genera un script SQL que cree una tabla **módulos** con algunos registros con los nombres de los módulos que estás estudiando
3. Crear un contenedor con **Adminer** o con **phpMyAdmin** que se pueda conectar al contenedor de la BD
4. Desde la interfaz gráfica elegida, conéctate a la BD con tu usuario personal, ejecuta el script con los datos de los módulos y muestra la BD y la tabla creados
5. **Elige** entre estas dos opciones:
 - Instala la extensión **Disk Usage**, muestra el espacio ocupado, borra algo...
 - Instala la extensión **Resource usage** y muestra su salida cuando estén activos los contenedores.

Entregar, al menos, las siguientes capturas de pantalla con las explicaciones correspondientes, para resolver cada apartado:

- Contenedores creados y en ejecución
- Parámetros de la red bridge utilizada: IPs

- Acceso a la BD a través de la interfaz web de Adminer o de phpMyAdmin con tu usuario
- BD creada y los datos de la tabla de módulos
- Borrar los contenedores, la red y los volúmenes utilizados

Ejercicio 2 - Docker Compose

Este ejercicio se puede resolver utilizando comandos, o Docker Desktop, o combinando ambos

Explorar la imagen de la aplicación `FileBrowser` en este repositorio en GitHub: <https://hub.docker.com/r/hurlenko/filebrowser>

El usuario por defecto es `admin` con contraseña 'admin'

Escribir un fichero `compose.yaml` para desplegarla. Los datos se pueden guardar utilizando volúmenes o utilizando bind-mount

Entregar, al menos, las siguientes capturas de pantalla y los comandos y/o operaciones con Docker Desktop empleados para resolver el ejercicio:

- Captura de pantalla y documento donde se vea el fichero `docker-compose.yaml` que has creado.
- Captura de pantalla donde se vean los volúmenes/carpetas donde se han almacenado los datos.
- Captura de pantalla donde se vea la aplicación funcionando, sube algún fichero, cambia el lenguaje a español...
- Explicar brevemente cómo funciona esta aplicación y qué hace.

Ejercicio 3 - imagen con Dockerfile -

Aplicación web

Para la realización de este ejercicio es necesario tener una cuenta en Docker Hub.

Este ejercicio se puede resolver utilizando comandos, o Docker Desktop, o combinando ambos

Necesitamos un fichero **Dockerfile** que automatice las siguientes operaciones para crear una imagen que contenga un servidor con un sitio web y un script php. Características de la imagen:

- Usa un contenedor que ejecute una instancia de la imagen `php:7.4-apache`, que se llame `ejercicio3` y que sea accesible desde un navegador en el puerto 8000.
- Coloca en el directorio raíz del servicio web (`/var/www/html`) un "sitio web" donde figure tu nombre - el sitio deberá tener al menos un archivo `index.html` sencillo y un archivo `.css`
- Coloca en ese mismo directorio raíz el siguiente script `php`, llámalo `fecha.php`

```
<?php
    setlocale(LC_TIME, "es_ES.UTF-8");
    $mes_actual = strftime("%B");
    $fecha_actual = date("d/m/Y");
    $hora_actual = date("H:i:s");
    echo "<h1>Información</h1>";
    echo "<p>Hoy es $fecha_actual</p>";
    echo "<p>El mes es: <strong>$mes_actual</strong></p>";
    echo "<p>Hora: $hora_actual</p>";
?>
```

- Ver la salida del script `fecha.php` y de la página `index.html` en el navegador

Una vez creada la imagen, súbela a tu cuenta de Docker Hub

- Borra la imagen de tu Docker local
- Baja ('pull') de tu cuenta la imagen que acabas de subir
- Muestra las imágenes que tienes
- Ejecuta un contenedor usando esa imagen

Deberás entregar, al menos, las siguientes capturas de pantalla, los comandos empleados y/o operaciones con Docker Desktop para resolver cada apartado:

- creación inicial del contenedor - documenta los pasos hasta el borrado del mismo
- bloque de código con el Dockerfile
- creación de la nueva imagen.
- subida de la imagen a tu cuenta de Docker Hub.
- operación de 'pull' de la imagen de Docker Hub
- creación de un nuevo contenedor con esa imagen y su ejecución. Cambia el puerto del contenedor, por ejemplo, `- p 1234:80`
- el acceso al navegador con la página `html` y con el script `php`

Videoclip

Realiza un videoclip con el siguiente contenido:

- Preséntate, debe verse tu cara
- Entra en GitHub y muestra tu repositorio
- Escoge uno de los tres ejercicios y resuelve alguno de los puntos solicitados, por ejemplo, del ejercicio 1, muestra que se están ejecutando los dos contenedores y conéctate a la BD utilizando tu usuario personal

Calificación

| Apartado | Puntuación |
|--|------------|
| Documentación <ul style="list-style-type: none">- se documenta cada ejercicio en Markdown, se entrega el .pdf.- los documentos contienen título, nombre autor/a, tabla de contenidos, detalle de la resolución de cada ejercicio con comandos, imágenes, etc | 1 |
| Repositorio <ul style="list-style-type: none">- se puede acceder al repositorio en GitHub- la estructura de carpetas es correcta, hay una carpeta para cada ejercicio- se ha hecho al menos una rama por ejercicio y están todas fusionadas en la rama principal- se pueden ver los ficheros .md con sus imágenes y los .pdf correspondientes- en los ejercicios donde se necesita usar archivos (html, yaml, etc.), se han subido correctamente | 1,25 |
| VideoClip <ul style="list-style-type: none">- se aprecia la identidad de la persona- tiene una duración correcta, entre 3 y 5 minutos- explica todo lo que se solicita | 1 |
| Ejercicios <ul style="list-style-type: none">- cada ejercicio resuelto correctamente completando todos los apartados que se solicitan : 2,25 puntos | 6,75 |