

Tarea Evaluable Docker

Módulo Despliegue de Aplicaciones Web - CIFP La Laboral

Tarea Evaluable Docker

Instrucciones

Ejercicio 1 - Servidor de base de datos

Ejercicio 2 - Portainer

Ejercicio 3 - contenedores en red: Adminer y MariaDB

Ejercicio 4 - Docker Compose

Ejercicio 5 - imagen con Dockerfile - Aplicación web

Instrucciones

- Estos ejercicios se realizarán en grupos de 2 personas
- Se debe crear un repositorio GitHub donde una persona será la dueña del repositorio (ejercerá labores de jefe/a del proyecto), y la otra serán colaboradora
 - Se debe trabajar con ramas
 - En los diferentes **commits** debe verse la aportación de los miembros de cada equipo a la tarea
 - El jefe/a del equipo asignará las tareas y marcará los plazos de entrega para cada miembro del equipo
 - En uno de los documentos del repositorio, debe aparecer dicha asignación de tareas y plazos, el tanto por ciento de cumplimiento de la tarea, el plazo estimado y el plazo en que se fue finalizando cada apartado
 - Cada grupo expondrá en clase su experiencia en este trabajo:
 - resultado final, mostrando el repositorio final
 - mayores dificultades encontradas
 - lo mejor del trabajo en grupo
 - lo peor del trabajo en grupo
 - duración máxima de cada exposición 5 minutos
 - *La exposición de la tarea será el 10% de la nota*
- Se creará una carpeta para cada ejercicio entregable, debe quedar claro qué ejercicio/s contiene cada una
- En la carpeta estará el documento/s **.md** que resuelve el ejercicio, y las imágenes necesarias
- En los ejercicios que llevan código (por ejemplo, Dockerfile, Compose), debe estar también en el repositorio el fichero/s utilizado
- Se entregará un documento **.pdf** con los nombres de los componentes del grupo y la URL del repositorio donde está la resolución de los ejercicios

- Se valorará la claridad en la organización del repositorio y la completitud de la documentación de las soluciones a los ejercicios

Ejercicio 1 - Servidor de base de datos

- Arrancar un contenedor que se llame `bbdd` y que ejecute una instancia de la imagen `mariadb` para que sea accesible desde el puerto 3306.
- Antes de arrancarlo visitar la página del contenedor en [Docker Hub](#) y establecer las variables de entorno necesarias para que:
 - La contraseña de root sea `root`.
 - Crear una base de datos automáticamente al arrancar que se llame `prueba`.
 - Crear el usuario `invitado` con la contraseña `invitado`.

Entregar un documento con las siguientes capturas de pantalla y los comandos empleados para resolver cada apartado:

- Captura de pantalla y documento que desde el navegador muestre el fichero `index.html`.
- Captura de pantalla y documento que desde un navegador muestre la salida del script `mes.php`.
- Captura de pantalla y documento donde se vea el tamaño del contenedor `web` después de crear los dos ficheros.
- Captura de pantalla y documento donde desde un cliente de base de datos (instalado en tu ordenador) se pueda observar que hemos podido conectarnos al servidor de base de datos con el usuario creado y que se ha creado la base de datos `prueba` (`show databases`). El acceso se debe realizar desde el ordenador que tenéis instalado docker, no hay que acceder desde dentro del contenedor, es decir, no usar `docker exec`.
- Captura de pantalla y documento donde se comprueba que no se puede borrar la imagen `mariadb` mientras el contenedor `bbdd` está creado.

Ejercicio 2 - Portainer

Utiliza una imagen de 'Portainer' para gestionar Docker.

Documenta la aplicación, su puesta en funcionamiento y realiza capturas de varias operaciones, por ejemplo:

- Muestra contenedores activos, para un contenedor, borra un contenedor
- Muestra alguna operación con redes Docker
- Muestra alguna operación con volúmenes Docker

Ejercicio 3 - contenedores en red: Adminer y MariaDB

1. Crea una red bridge `redbd`
2. Crea un contenedor con una imagen de `mariaDB` que estará en la red `redbd`. Este contenedor se ejecutará en segundo plano, y será accesible a través del puerto 3306. (Es necesario definir la contraseña del usuario `root` y un volumen de datos persistente)
3. Crear un contenedor con `Adminer` que se pueda conectar al contenedor de la BD
4. Comprobar que el contenedor Adminer puede conectar con el contenedor mysql abriendo un navegador web y accediendo a la URL: <http://localhost:8080>

Entregar los siguientes Captura de pantalla y documento s y los comandos empleados para resolver cada apartado:

- Captura de pantalla y documento donde se vean los contenedores creados y en ejecución
- Captura de pantalla y documento donde se vea el acceso a la BD a través de la interfaz web de Adminer
- Captura de pantalla y documento donde se vea la creación de una BD con la interfaz web Adminer
- Captura de pantalla y documento donde se entre a la consola del servidor web en modo texto y se compruebe que se ha creado la BD
- Borrar los contenedores la red y los volúmenes utilizados

Ejercicio 4 - Docker Compose

Desplegar la aplicación *cmatrix* utilizando docker-compose.

Entregar las siguientes capturas de pantalla y los comandos empleados para resolver el ejercicio:

- Captura de pantalla y documento donde se vea el fichero `docker-compose.yaml`.
- Captura de pantalla y documento donde se vea la aplicación funcionando. Se valorará conseguir el efecto *Greenrain*.
- Explicar brevemente cómo funciona esta aplicación.

Si esta aplicación no funciona, puedes usar cualquier otra que NO se haya usado en clase

Ejercicio 5 - imagen con Dockerfile - Aplicación web

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

- Arranca un contenedor que ejecute una instancia de la imagen `php:7.4-apache`, que se llame `web` 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 el nombre de los componentes del grupo - el sitio deberá tener al menos un archivo `index.html` y un archivo `.css`.
- Coloca en ese mismo directorio raíz un archivo llamado `mes.php` que muestre el nombre del mes actual. Ver la salida del script en el navegador
- Borrar el contenedor
- Automatizar estas operaciones creando un fichero Dockerfile
- Subir la imagen a la cuenta de Docker Hub

Deberás entregar los siguientes capturas de pantalla y los comandos empleados para resolver cada apartado:

1. Creación inicial del contenedor - documenta los pasos hasta el borrado del mismo
2. Bloque de código con el Dockerfile
3. Captura de pantalla y documento donde se vea el comando que crea la nueva imagen.
4. Captura de pantalla y documento donde se vea la imagen subida a tu cuenta de Docker Hub.
5. Captura de pantalla y documento donde se vea la bajada de la imagen - por parte de otra persona del grupo - y la creación de un contenedor.
6. Captura de pantalla y documento donde se ve el acceso al navegador con el sitio servido

Se puede consultar la rúbrica detallada en el aula virtual