

Ejercicios Docker - entregables

Ejercicios Docker - entregables

Instrucciones

Ejercicio - inicial

Ejercicio - trabajo con imágenes

 Servidor web

 Servidor de base de datos

Ejercicio - almacenamiento

 Bind mount para compartir datos

Ejercicio - redes

 Despliegue de contenedores en red: Adminer y MariaDB

Ejercicio - crea una imagen con Dockerfile

 Crear una imagen con un servidor web que sirva un sitio web

Instrucciones

- Estos ejercicios se realizarán en grupos de 3 personas
- Se debe crear un repositorio GitHub donde una persona será la dueña del repositorio (ejercerá labores de jefe/a del proyecto), y el resto del equipo colaboradores
 - 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 15 minutos
- 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), 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 - inicial

Crear un contenedor demonio a partir de la imagen `nginx`, el contenedor se debe llamar `servidor_web` y se debe acceder a él utilizando el puerto 8181 del ordenador donde tengas instalado docker.

Entrega un documento con los siguientes pantallazos, y los comandos empleados para resolver cada apartado:

1. Pantallazo donde se vea la creación del contenedor y podamos comprobar que el contenedor está funcionando.
2. Pantallazo donde se vea el acceso al servidor web utilizando un navegador web (recuerda que tienes que acceder a la ip del ordenador donde tengas instalado docker)
3. Pantallazo donde se vean las imágenes que tienes en tu registro local.
4. Pantallazo donde se vea cómo se elimina el contenedor (recuerda que antes debe estar parado el contenedor).

Ejercicio - trabajo con imágenes

Servidor web

- 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.
- Colocar en el directorio raíz del servicio web (`/var/www/html`) de dicho contenedor un fichero llamado `index.html` con el siguiente contenido:

```
<h1>HOLA SOY XXXXXXXXXXXXXXXX</h1>
```

Deberás sustituir XXXXXXXXXXXX por tu nombre y tus apellidos.

- Colocar 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

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 3336.
- 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 los siguientes pantallazos, y los comandos empleados para resolver cada apartado:

- Pantallazo que desde el navegador muestre el fichero `index.html`.
- Pantallazo que desde un navegador muestre la salida del script `mes.php`.
- Pantallazo donde se vea el tamaño del contenedor `web` después de crear los dos ficheros.
- Pantallazo 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`.
- Pantallazo donde se comprueba que no se puede borrar la imagen `mariadb` mientras el contenedor `bbdd` está creado.

Ejercicio - almacenamiento

Bind mount para compartir datos

1. Crea una carpeta llamada `saludo` y dentro de ella crea un fichero llamado `index.html` con el siguiente contenido (Deberás sustituir ese XXXXXx por tu nombre.):

```
<h1>HOLA SOY XXXXXX</h1>
```

2. Una vez hecho esto arrancar dos contenedores basados en la imagen `php:7.4-apache` que hagan un bind mount de la carpeta `saludo` en la carpeta `/var/www/html` del contenedor. Uno de ellos vamos a acceder con el puerto 8181 y el otro con el 8282. Y su nombres serán `c1` y `c2`.
3. Modifica el contenido del fichero `~/saludo/index.html`.
4. Comprueba que puedes seguir accediendo a los contenedores, sin necesidad de reiniciarlos.
5. Borra los contenedores utilizados.

Entregar un documento con los siguientes pantallazos y los comandos empleados para resolver cada apartado:

- Pantallazo con la orden correspondiente para arrancar el contenedor c1 (puerto 8181) realizando el bind mount solicitado.
- Pantallazo con la orden correspondiente para arrancar el contenedor c2 (puerto 8282) realizando el bind mount solicitado.
- Pantallazo donde se pueda apreciar que accediendo a `c1` se puede ver el contenido de `index.html`.
- Pantallazo donde se pueda apreciar que accediendo a `c2` se puede ver el contenido de `index.html`.
- Otros dos pantallazos donde se vea el acceso al fichero `index.html` después de modificarlo.
- Borrar los dos contenedores. Mostrar que se han borrado.

Ejercicio - redes

Despliegue de 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 pantallazos y los comandos empleados para resolver cada apartado:

- Pantallazos donde se vean los contenedores creados y en ejecución
- Pantallazo donde se vea el acceso a la BD a través de la interfaz web de Adminer
- Pantallazo donde se vea la creación de una BD con la interfaz web Adminer
- Pantallazo 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 - crea una imagen con Dockerfile

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

Crear una imagen con un servidor web que sirva un sitio web

- Basar la imagen en `nginx` o `apache`
- Desplegar una plantilla, o un trabajo de clase, que tenga, al menos, un `index.html` y una carpeta para estilos, imágenes, etc.

Deberás entregar los siguientes pantallazos y los comandos empleados para resolver cada apartado:

1. Pantallazo/bloque de código con el Dockerfile
2. Pantallazo donde se vea el comando que crea la nueva imagen.
3. Pantallazo donde se vea la imagen subida a tu cuenta de Docker Hub.
4. Pantallazo donde se vea la bajada de la imagen - por parte de otra persona del grupo - y la creación de un contenedor.
5. Pantallazo donde se ve el acceso al navegador con el sitio servido