

Actividad evaluable 3 - Git y Docker

Actividad evaluable 3 - Git y Docker

Instrucciones

Ejercicio 1 - trabajo con imágenes

Servidor web

Servidor de base de datos

Entregar:

Ejercicio 2 - almacenamiento

Bind mount para compartir datos

Entregar:

Ejercicio 3 - redes

Despliegue de contenedores en red: Adminer y MariaDB

Entregar:

Ejercicio 4 - imagen con Dockerfile

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

Entregar:

Videoclip

Calificación

Instrucciones

- Se debe crear un repositorio GitHub para contener las soluciones a los ejercicios:
 - Debe haber una **carpeta** con la solución de cada uno de los ejercicios
 - Para solucionar cada ejercicio, se creará una rama, se irán subiendo las imágenes y las distintas versiones del fichero **.md** con la solución. Una vez finalizada la solución de esa parte del ejercicio, se sube el **.pdf**, se hace un commit en la rama y se hace un 'merge' con la rama principal
- En los ejercicios que llevan algún fichero con código (por ejemplo, Dockerfile, html, php...), deben estar también en la carpeta correspondiente al ejercicio el fichero/s utilizado/s
- Se subirá al aula virtual un documento **.pdf** con el nombre del autor/a, la URL del repositorio donde está la resolución de los ejercicios y el enlace al video de la tarea
- Se valorará la claridad en la organización del repositorio y la completitud de la documentación de las soluciones a los ejercicios, incluyendo los comandos y pantallazos solicitados
- Se puede ver un ejemplo en este repositorio: <https://github.com/inesAlumnaDaw/tarea-docker>

Ejercicio 1 - 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`
- Antes de arrancarlo, visita la página en [Docker Hub](#)
- Establece 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 de la consola de la Base de Datos donde 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`).
- Pantallazo donde se comprueba que no se puede borrar la imagen `mariadb` mientras el contenedor `bbdd` está creado.
- Pantallazo donde se vean las imágenes que tienes en tu registro local.
- Pantallazo donde se vea cómo se eliminan los contenedores utilizados.

Ejercicio 2 - 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 XXXXX por tu nombre.):

```
<h1>HOLA SOY XXXXX</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 3 - 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. Crea un contenedor con `Adminer` que se pueda conectar al contenedor de la BD
4. Comprueba 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 4 - 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 proyecto tuyo, que tenga, al menos, un `index.html` y una carpeta para estilos, imágenes, etc.

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 ve el acceso al navegador con el sitio servido
4. Pantallazo donde se vea la imagen subida a tu cuenta de Docker Hub.

Videoclip

Realiza un videoclip donde se vea tu cara, con el siguiente contenido:

- Presentación
- Muestra el acceso TU repositorio GitHub con las soluciones de los ejercicios. Muestra los commits y ramas que has ido haciendo (debe quedar clara la identidad del propietario/a del repositorio)
- Aprovechando tu solución del ejercicio de **REDES**, ejecuta 'en directo' los comandos para resolver los apartados 1 a 4, de forma que se vea su funcionamiento
- Duración máxima del video: 5 minutos

Calificación

Apartado	Calificación
Documentación correcta, con todos los puntos solicitados y bien formateada	1
Repositorio correcto, con las carpetas y ramas y todo el contenido que se solicita	3
Ejercicio imágenes y ejercicio almacenamiento	1 + 1
Ejercicio redes y ejercicio Dockerfile	1,5 + 1,5
Videoclip - indispensable para la calificación de la tarea - se tendrá en cuenta que se aprecie la identidad de la persona, que se expliquen los puntos pedidos, y que la duración sea correcta	1

Si no se entrega videoclip la nota de la tarea será 0