

GUIÓN DE LA ACTIVIDAD AEV7:

Título

Contenedor personalizado

Objetivos

- Entender las instrucciones descriptivas de un fichero de configuración Dockerfile, para poder integrar una aplicación propia en el proceso de generación de una imagen.
- Entender el concepto de adición incremental de capas (delta).
- Conocer los pasos del proceso de creación de una imagen a partir de un fichero de configuración Dockerfile.
- Conocer el proceso de integración de una imagen en un repositorio de un registro público.
- Conocer los aspectos a tener en cuenta durante el proceso de creación y puesta en marcha de un contenedor, en base a una imagen.
- Conocer el proceso de parada e inicio de un contenedor existente.

Temporalización

Se estima una dedicación de **5 horas**, teniendo en cuenta que habrá que revisar los recursos publicados en Florida Oberta para poder realizar la actividad.

Proceso de desarrollo

1. Se van a proponer una serie de ejercicios o pasos en los que habrá que realizar una o varias tareas o cuestiones específicas.
2. Conforme se realice cada tarea, se realizará una explicación oportuna en un documento junto con capturas de pantalla. De modo que se argumente, explique y confirme que el paso ha sido realizado y cómo.
3. Entregar un documento PDF, debidamente identificado, que incluya cada enunciado con la respuesta correspondiente, a través de Florida Oberta.

Evaluación

Cada uno de los ejercicios dispondrá de una valoración en puntos, en función de su dificultad o esfuerzo requerido. En total sumarán 10 puntos. Como norma, cada ejercicio se valorará del siguiente modo:

- Errores graves: la respuesta no corresponde a lo solicitado. El ejercicio puede llegar a sumar 0 puntos.
- Errores leves:
 - Ejercicio parcialmente incorrecto, resta entre 25% - 100%.
 - Cada fallo leve resta entre 10% - 50%.

Recursos

Puestos a disposición del alumno en el curso correspondiente del campus virtual Florida Oberta.

Detalle de la actividad

1. Obtención de la aplicación. (1 punto)

Vamos a basar la ejecución de esta actividad en una pequeña demo, propuesta por un especialista en Cloud Computing, particularmente en entornos de contenedores. Su nombre es Nigel Poulton, ha escrito varios libros sobre el tema y es instructor de cursos online de la plataforma “Pluralsight”, además de haber participado de forma activa en el desarrollo y difusión global de este tipo de tecnologías.

Usaremos como base, los ficheros del repositorio psweb de la cuenta NigelPoulton en GitHub, es decir: <https://github.com/nigelpoulton/psweb>

Accede al repositorio y descarga el contenido en una carpeta local. También podríamos clonarlo, pero como posteriormente no vamos a integrar nuestros cambios en el repositorio, con descargar los ficheros es suficiente en este caso.

2. Personalización de la aplicación. (1 punto)

Una vez descomprimida la descarga, hacemos una copia de trabajo del contenido completo de “psweb-master” en otra carpeta local. Observamos que disponemos de un conjunto de directorios y ficheros que compondrán una pequeña aplicación. Será la aplicación que integraremos en un contenedor.

En el directorio raíz de la aplicación dispondremos, entre otros, de un fichero Dockerfile. Para poder visualizar el contenido del fichero, puedes usar el editor “Visual Studio Code”. El fichero Dockerfile es un poco especial, puesto que no tiene extensión.

En este caso, el fichero Dockerfile nos viene dado (se explica su estructura en el recurso didáctico correspondiente, en el curso de Florida Oberta).

Modifica el fichero Dockerfile. Concretamente, debes indicar tu correo electrónico de contacto en la etiqueta **LABEL maintener**.

Respecto al resto de los ficheros descargados en nuestra carpeta local y que formarán nuestra aplicación web, accede al fichero “**home.pug**” de la carpeta “**views**”. La aplicación es un desarrollo del tipo “hola mundo”, que muestra una sencilla página como resultado. Este fichero define el contenido de la página principal que se mostrará cuando arranque el contenedor.

Edita el fichero, puedes hacerlo también con “Visual Studio Code”, y personalízalo. Modifica los textos y los enlaces, poniendo unos personalizados por ti, los que tú quieras. Esto hará que tu contenedor sea único.

Llegados a este punto y habiendo guardado los cambios, ya tenemos preparada nuestra aplicación personalizada, para poder “hacerle una foto” y meterla en un contenedor.

3. Generación de la imagen. (2 puntos)

Genera una imagen Docker que contenga nuestra aplicación, a partir de las especificaciones descritas en el fichero Dockerfile. Confirma mediante la línea de comandos que la imagen se ha generado (opcionalmente, también mediante Docker Desktop).

4. Integración de la imagen en Docker Hub. (1 puntos)

Guarda la imagen generada en Docker Hub, mediante el uso de comandos. Confirma que existe en el repositorio.

5. Ejecución de un contenedor en base a la imagen generada. (3 puntos)

Crea un contenedor basado en la imagen generada. El contenedor escuchará las peticiones por el puerto 8080, que a su vez hayan entrado previamente a través del host anfitrión (máquina física) mediante el puerto 8000. Confirma que se muestra la página web correspondiente. Detén el contenedor, confirma que ahora no se muestra la web y posteriormente vuelve a arrancarlo. Todo mediante el uso de comandos (opcional y adicionalmente, puedes hacerlo también mediante Docker Desktop).

6. Ejecución de un contenedor con volumen en base a la imagen generada. (2 puntos)

Crea un nuevo contenedor basado en la imagen generada, pero en este caso con un volumen. De forma que queden mapeadas las carpetas denominadas “views” del contenedor y del sistema de archivos del anfitrión o máquina física. Confirma que si modificas el fichero “home.pug” desde el anfitrión, éste se modifica en el contenedor y, en consecuencia, los cambios se verán reflejados en la página web correspondiente. Confirma también que, si lo modificas desde el contenedor, se modifica en el sistema de archivos del anfitrión.