

Imágenes

Las imágenes son la base de Docker. Nuestros contenedores se iniciarán a partir de ellas. Como se indicó en la introducción, es una plantilla de solo lectura, que se crea incorporando los requisitos necesarios para cumplir el objetivo para el cual fue creada.

Por ejemplo, si estamos creando un proyecto con PHP, incorporará el intérprete del lenguaje de PHP. Si es una página web, incorporará el servidor web (*apache*, *nginx*, etc.).

En este apartado vamos a ver el manejo de imágenes con Docker. Para ello seguiremos este documento.

Imágenes Docker

Para ir recordando los distintos comandos que vayamos usando imprime y ve marcando los comandos que uses en este CheatSheet:

Cheatsheet

Una vez finalices la parte teórica prueba a realizar estas dos prácticas para afianzar conocimientos:

Caso práctico 01 - Creando una imagen Ubuntu con nano

Caso práctico 01 - Creando una imagen Ubuntu con nano

Caso práctico 02 - Apache 2 con PHP desde Alpine

Caso práctico 02 - Apache 2 con PHP desde Alpine

Warning

Esta práctica despliega un servidor web APACHE con Php. Todavía no hemos visto el tema de servidores web así que no te preocupes por los detalles referentes al servidor web. Lo importante de esta práctica es ver cómo usa los distintos comandos en el Dockerfile ADD, RUN, EXPOSE... Lo que va detrás no es necesario que lo entiendas ahora

En esta práctica, en el punto 2 veréis que el Dockerfile contiene la línea "ADD ./start.sh /start.sh". Tenéis que crear en el mismo directorio donde está el Dockerfile un archivo `start.sh` con este contenido:

```
#!/bin/sh
#Lanzamos servicio Apache2 en segundo plano
/usr/sbin/httpd -D FOREGROUND
```

En el punto 3. Probando la imagen te propone lanzar la aplicación sobre el puerto 80 de nuestra máquina. Si ese puerto ya estuviera usado por otra aplicación te podría dar problemas. En ese caso puedes lanzarlo sobre el puerto 8080 con el comando:

```
docker run -d -p 8080:80 alpineapache
```

En este caso para acceder después desde el navegador tendrías que usar:

```
http://localhost:8080
```

Recuerda que si estuvieras trabajando sobre una máquina virtual en AWS Academy, deberás abrir el puerto 8080 y sustituir localhost por la IP pública de esa máquina virtual.