

# DESPLIEGUE DE APLICACIONES WEB

## Tema 1 : Implantación de Arquitecturas Web

Docker permite probar y usar cualquier servicio y cualquier aplicación sin tener que instalar nada en el sistema operativo.

Para la ejecución de los contenedores necesitaremos el comando **docker pull nombre\_imagen:version** , la cual descargará desde el repositorio una imagen con la versión indicada.

Para el arranque de un contenedor ya creado anteriormente y apagado se necesitará el comando **docker run [opción] nombre\_imagen** , este comando tiene diferentes opciones:

- **-d o --detach** para ejecutar un contenedor (con un servicio) en background.
- **-e o --env** para establecer variables de entorno en la ejecución del contenedor.
- **-h o --hostname** para establecer el nombre de red para el contenedor.
- **--help** para obtener ayuda de las opciones de docker.
- **--interactive o -i** para mantener la STDIN abierta en el contenedor.
- **--ip** si quiero darle una ip concreta al contenedor.
- **--name** para darle nombre al contenedor.
- **--net o --network** para conectar el contenedor a una red determinada.
- **-p o --publish** para conectar puertos del contenedor con los de nuestro host.
- **--restart** que permite reiniciar un contenedor si este se "cae" por cualquier motivo.
- **--rm** que destruye el contenedor al pararlo.
- **--tty o -t** para que el contenedor nos permita un acceso a un terminal para ejecutar órdenes .
- **--user o -u** para establecer el usuario con el que vamos a ejecutar el contenedor.
- **--volume o -v** para montar un bind mount o un volumen en nuestro contenedor.
- **--workdir o -w** para establecer el directorio de trabajo en un contenedor.

Hay veces que los contenedores se paran y ya no se puede acceder a ellos ni iniciarlos de nuevo, para que no pase esto se tiene que utilizar siempre el **flag it** al ejecutar el **docker run**

Para inicializar un contenedor se hará con el comando **docker run nombre\_imagen**

Usar el flag **-d** para que el servicio se ejecute en modo background o detach, si esto no lo hacemos se bloqueará el terminal mostrando el log del servicio

Usar el flag **-p** para que el servicio sea accesible desde el exterior, **-p (puerto\_en\_host:puerto\_en\_contenedor)**

Usar el flag **-e** para comprobar y definir si es necesario las variables de entorno que puede tener el contenedor, **-e nombre\_variable = valor**

Para asignar el nombre a un contenedor necesitaremos poner el **--name** cuando usamos el comando **docker run** , **docker run -d --name nombre\_a\_poner**

Para ejecutar ordenes en contenedores es necesario que el contenedor esté en ejecución

***docker exec [opciones] nombre\_contenedor orden [argumentos]***

Algunas opciones mas importantes pueden ser:

***-it*** para tener interactividad con el contenedor ejecutando un shell

***-u*** o ***-user*** si quiero ejecutar la orden como si fuera un usuario distinto del de root

***-w*** o ***-workdir*** si quiero ejecutar la orden desde un directorio concreto

Para permitir copiar ficheros utilizaremos el comando de docker cp desde el sistema al contenedor y desde el contenedor a mi sistema

***docker cp [fichero]***

Para obtener informacion de los contenedores ya arrancados necesitaremos el comando ***docker ps***, el cual te da el estado, la imagen, el tamaño, el Entrypoint, el nombre, cuando fue creado, las redirrecciones de puertos.

Para una información detallada del contenedor utilizaremos el comando ***docker inspect***

Con el comando ***docker logs*** da informacion de lo que está pasando en el contenedor, se pueden utilizar para contenedores parados y en ejecución.

Para la gestión de contenedores tenemos varios comandos, ***docker stop*** para parar el contenedor, ***docker rm*** para borrar el contenedor, ***docker start*** para iniciar un contenedor que estaba parado previamente, ***docker restart*** para reiniciar un contenedor que previamente ya está en ejecución