

Práctica de Docker Avanzado en Terraformadores de Venus

En **Terraformadores de Venus** trabajamos cada día con tecnologías que permiten desplegar servicios de forma rápida, escalable y sin errores de compatibilidad. Entre ellas, **Docker** es una de las más importantes: nos permite empaquetar aplicaciones en contenedores listos para ejecutarse en cualquier entorno.

En esta práctica vas a realizar una serie de ejercicios para afianzar los conceptos clave de Docker: **contenedores en segundo plano, uso de volúmenes, construcción de imágenes personalizadas, orquestación con Docker Compose y despliegue de aplicaciones reales.**

La secuencia de ejercicios está pensada para que paso a paso adquieras soltura en el uso de Docker, siempre con ejemplos aplicables a los proyectos de Terraformadores.

Ejercicio 1: Servidor Nginx en modo demonio

Terraformadores necesita desplegar un servicio web ligero para mostrar documentación interna.

- Crea un contenedor en **segundo plano (modo demonio)** utilizando la imagen oficial de **nginx**.
- Verifica desde el navegador de tu máquina que el servidor está funcionando correctamente.
- Investiga cómo **ver los logs** del contenedor y muéstralos en tu entrega.

✧ *Este ejercicio te permitirá comprobar lo fácil que resulta levantar servicios web con Docker y monitorizar su actividad.*

Pistas

- Para arrancar un contenedor en segundo plano usa:

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

- Para ver los logs del contenedor:

```
docker logs <ID_contenedor>
```

Ejercicio 2: Uso de volúmenes en Nginx

En Terraformadores no basta con levantar contenedores: necesitamos **persistencia de datos** y poder modificar contenidos fácilmente.

- Crea un contenedor de **nginx** que utilice un volumen para servir los archivos de la página web.

- Modifica un fichero HTML en el **host** y comprueba que el cambio se refleja inmediatamente en el contenedor.
- Explica la diferencia entre un **volumen nombrado** y un **bind mount**, e indica cuál has usado en tu prueba.

✦ Este ejercicio te ayudará a entender cómo se maneja la persistencia y la compartición de datos entre contenedores y host.

🔍 Pistas

- Un **bind mount** se define con `-v ruta_host:ruta_contenedor`.
- Por ejemplo, si en tu host tienes un directorio `./miweb` con un `index.html`, puedes montarlo así:

```
docker run -d -p 8080:80 -v ./miweb:/usr/share/nginx/html nginx
```

- Cualquier cambio en `miweb/index.html` se reflejará al instante.

🔧 Ejercicio 3: Construcción de una imagen con Dockerfile

Terraformadores necesita personalizar algunos de sus servicios para que no dependan solo de imágenes externas.

- Crea un **Dockerfile** que parta de la imagen oficial de **nginx** y copie dentro un `index.html` personalizado.
- Construye tu propia imagen con un nombre reconocible.
- Lanza un contenedor a partir de tu nueva imagen y comprueba que sirve tu página personalizada.

✦ Este ejercicio te mostrará cómo generar imágenes personalizadas y reutilizables, clave en los despliegues internos de Terraformadores.

🔍 Pistas

- Usa en el **Dockerfile**:

```
FROM nginx
COPY index.html /usr/share/nginx/html/
```

- Construir la imagen:

```
docker build -t mi-nginx-personalizado .
```

- Ejecutar un contenedor desde tu nueva imagen:

```
docker run -d -p 8080:80 mi-nginx-personalizado
```

⚙️ Ejercicio 4: Orquestación básica con Docker Compose

En Terraformadores necesitamos coordinar varios servicios a la vez, y hacerlo contenedor por contenedor resulta ineficiente. Para ello utilizamos **Docker Compose**, que permite levantar entornos completos con un solo fichero.

- Crea un fichero `docker-compose.yml` con dos servicios:
 - Un servicio **nginx** expuesto en el puerto 8080.
 - Un servicio **goaccess** (analizador de logs) que se conecte al contenedor de nginx para monitorizar en tiempo real sus registros de acceso.
- Ambos servicios deben estar en la misma red interna definida en el fichero Compose.
- Arranca los servicios con `docker-compose up -d`.
- Comprueba que puedes acceder a la web servida por nginx y que goaccess está procesando sus logs.

✂️ *Este ejercicio no despliega la típica aplicación web con base de datos, sino que te muestra cómo levantar un pequeño ecosistema de servicios colaborativos, muy habitual en entornos profesionales como Terraformadores.*

🔍 Pistas

- La estructura básica de un `docker-compose.yml` es:


```
version: '3'
services:
  servicio1:
    image: ...
  servicio2:
    image: ...
```

- Para conectar servicios entre sí, basta con declararlos dentro del mismo fichero Compose (la red se crea automáticamente).
- Comandos útiles:
 - Levantar servicios: `docker-compose up -d`
 - Listar servicios: `docker-compose ps`
 - Ver logs: `docker-compose logs -f`

Ejercicio 5: Despliegue de Nextcloud

Terraformadores necesita un sistema de almacenamiento en la nube para compartir documentación entre los distintos equipos de trabajo. Para ello vamos a desplegar **Nextcloud** dentro de un contenedor Docker.

- Revisa la documentación oficial de Nextcloud en **Docker Hub**.
- Crea un contenedor que despliegue la aplicación Nextcloud.
- Personaliza la configuración para que use una base de datos **sqlite** con un nombre de fichero definido por ti.
- Accede desde el navegador y realiza el proceso de configuración inicial.

 Con este ejercicio practicarás el despliegue de una aplicación real de uso profesional utilizando Docker.

Pistas

- Busca en Docker Hub la imagen **nextcloud**.
- Puedes lanzar el contenedor con un volumen para persistir datos:

```
docker run -d -p 8080:80 -v ./nextcloud:/var/www/html nextcloud
```

- Durante la configuración inicial, podrás elegir **SQLite** como motor de base de datos y definir el nombre del fichero de datos.

Entrega

Debes entregar un documento en **Markdown** en tu repositorio de **GitLab**, que incluya:

- Una breve explicación de cada paso realizado.
- Capturas de pantalla que demuestren el funcionamiento de cada ejercicio.
- Una reflexión final sobre qué conceptos de Docker has reforzado en esta práctica.