

Práctica Guiada de Docker para Taller

Objetivo: Crear y gestionar un contenedor de Docker para entender los comandos básicos. Esta práctica está diseñada para que los participantes completen la actividad en menos de 10 minutos y se familiaricen con los comandos fundamentales de Docker.

Requisitos Previos

1. Tener Docker instalado en la máquina.
-

Parte 1: Práctica Guiada Básica (5 Minutos)

1. Verificar la instalación de Docker

Ejecuta este comando para asegurarte de que Docker está funcionando:

```
docker --version
```

2. Descargar una imagen de Docker

Vamos a descargar una imagen simple de Nginx (un servidor web):

```
docker pull nginx
```

3. Listar las imágenes descargadas

Verifica que la imagen de Nginx ha sido descargada correctamente:

```
docker images
```

4. Correr un contenedor usando la imagen de Nginx

Ahora vamos a iniciar un contenedor con Nginx para probar un servidor web:

```
# Sustituir el puerto 8080 por el puerto asignado en el taller
docker run -d -p 8080:80 --name mi_nginx nginx
```

- **Explicación de los parámetros:**

- `-d` : Ejecuta el contenedor en segundo plano.
- `-p 8080:80` : Mapea el puerto 80 del contenedor al puerto 8080 de la máquina local.
- `--name mi_nginx` : Asigna un nombre específico al contenedor.

5. Verificar que el contenedor está en ejecución

Lista los contenedores en ejecución para confirmar que el contenedor de Nginx está activo:

```
docker ps
```

6. Probar el servidor Nginx

Abre un navegador e ingresa a `https://tidpnube-b.rediris.es/TU_USUARIO/docker` .
Deberías ver la página de bienvenida de Nginx.

7. Detener y eliminar el contenedor

Para detener y eliminar el contenedor una vez completada la práctica, ejecuta:

```
docker stop mi_nginx  
docker rm mi_nginx
```

8. Limpiar las imágenes (No borrar antes de de hacer la parte 2 voluntaria)

Si quieres liberar espacio en tu máquina, puedes eliminar la imagen descargada:

```
docker rmi nginx
```

Parte 2: Práctica Extendida (5-10 Minutos)

1. Ver el estado de un contenedor en detalle

```
docker inspect mi_nginx
```

2. Ver logs del contenedor en tiempo real

```
docker logs -f mi_nginx
```

3. Entrar en el contenedor para explorar el sistema de archivos

```
docker exec -it mi_nginx /bin/bash
```

4. Ejecutar una pequeña modificación en el contenedor

Crea un archivo en `/usr/share/nginx/html/` y verifica el cambio en el navegador.

5. Detener y reiniciar el contenedor

```
docker start mi_nginx
docker stop mi_nginx
```

Parte 3: Docker Cheat Sheet

Comando	Descripción
<code>docker --version</code>	Muestra la versión de Docker instalada.
<code>docker pull <imagen></code>	Descarga una imagen del repositorio de Docker Hub.
<code>docker images</code>	Lista todas las imágenes descargadas en el sistema.
<code>docker run <imagen></code>	Ejecuta un contenedor usando una imagen específica.
<code>docker run -d -p <host>:<container> <imagen></code>	Ejecuta un contenedor en segundo plano y mapea puertos.
<code>docker ps</code>	Muestra todos los contenedores en ejecución.
<code>docker ps -a</code>	Muestra todos los contenedores (activos e inactivos).
<code>docker stop <nombre_contenedor></code>	Detiene un contenedor en ejecución.
<code>docker rm <nombre_contenedor></code>	Elimina un contenedor detenido.

<code>docker rmi <imagen></code>	Elimina una imagen de Docker del sistema.
<code>docker logs <nombre_contenedor></code>	Muestra los logs de un contenedor.
<code>docker exec -it <nombre_contenedor> <comando></code>	Ejecuta un comando dentro de un contenedor en ejecución.
<code>docker inspect <nombre_contenedor></code>	Muestra información detallada sobre el contenedor.
<code>docker logs -f <nombre_contenedor></code>	Sigue en tiempo real los logs del contenedor.
<code>docker exec -it <nombre_contenedor> bash</code>	Abre una terminal interactiva dentro del contenedor.
<code>docker start <nombre_contenedor></code>	Inicia un contenedor detenido.
<code>docker stop <nombre_contenedor></code>	Detiene un contenedor en ejecución.
<code>docker pause <nombre_contenedor></code>	Pausa un contenedor.
<code>docker unpause <nombre_contenedor></code>	Reactiva un contenedor pausado.
<code>docker stats</code>	Monitorea en tiempo real el uso de recursos de los contenedores.

////////////////////////////////////

¡Con esta práctica, ahora tienes una comprensión básica de cómo usar Docker para gestionar y operar contenedores!