

Máquinas o laboratorios vulnerables en Docker.

Requisitos Previos

- Tener Docker instalado en tu sistema Linux (Kali Linux, Ubuntu, etc.).
- El video recomienda conocer la IP del contenedor para pruebas: `hostname -I` [[05:07](#)].

Método 1: Creación Manual (Para principiantes en Docker)

Este método consiste en entrar a un contenedor base, configurarlo interactivamente y luego guardar los cambios.

1. Descargar y ejecutar la imagen base

```
# Descargar la imagen de Ubuntu (última versión)
```

```
docker pull ubuntu:latest
```

```
# Ejecutar el contenedor de forma interactiva
```

```
docker run -it ubuntu:latest /bin/bash
```

[[04:02](#)] - [[04:33](#)]

2. Instalar servicios y configurar la máquina

Dentro del contenedor, puedes instalar lo que necesites para tu laboratorio:

```
apt update
```

```
apt install apache2 open          # Instalar servidor Web y SSH
```

```
apt install vsftpd                 # Instalar servidor FTP
```

```
service apache2 start
```

```
service ssh start
```

```
service vsftpd start
```

[\[05:36\]](#) - [\[07:57\]](#)

3. Crear persistencia y "Snapshots"

Como los cambios en un contenedor se pierden si no se guardan, debes convertirlos en una imagen:

```
# Identificar el ID del contenedor desde otra terminal  
docker ps
```

```
# Crear la imagen (snapshot)  
docker commit [ID_CONTENEDOR] snapshot1:latest
```

[\[09:07\]](#) - [\[09:28\]](#)

4. Asegurar que los servicios arranquen automáticamente

Para que al exportar la máquina los servicios no estén detenidos, crea un archivo llamado Dockerfile:

```
Dockerfile  
FROM snapshot1:latest  
CMD service ssh start && service apache2 start && service vsftpd start && tail -f /dev/null
```

Nota: El comando tail -f /dev/null es vital para que el contenedor no se detenga inmediatamente [\[12:15\]](#).

Método 2: Usando un Dockerfile (Recomendado)

Este método automatiza todo el proceso de creación en un solo archivo de texto llamado Dockerfile.

1. Crear el archivo de configuración

Crea un archivo llamado Dockerfile con el siguiente contenido (ejemplo del video):

```
Dockerfile  
FROM debian:latest
```

```

# Actualizar e instalar servicios y herramientas
RUN apt update && apt upgrade -y && \
    apt install -y apache2 vsftpd ssh sudo python3

# Crear usuarios y contraseñas
RUN useradd -m bob && echo "bob:password1" | chpasswd
RUN useradd -m gladis && echo "gladis:chocolate" | chpasswd

# Configurar permisos SUDO (Escalada de privilegios)
RUN echo "bob ALL=(gladis) /usr/bin/python3" >> /etc/sudoers
RUN echo "gladis ALL=(root) /usr/bin/dpkg" >> /etc/sudoers

# Configurar binarios SUID (Opcional)
RUN chmod u+s /usr/bin/env

# Comando para iniciar servicios y mantener vivo el contenedor
CMD service apache2 start && service vsftpd start && tail -f /dev/null
[15:00] - [18:49]

```

2. Construir la imagen

```

docker build -t mi_maquina:latest .
[19:02]

```

Exportar la máquina terminada

Una vez que tu máquina esté lista y probada, puedes exportarla a un archivo .tar para compartirla:

```

# Guardar la imagen en un archivo comprimido
docker save -o mi_maquina.tar mi_maquina:latest
[13:40] - [20:39]

```

Este archivo .tar es el que puedes distribuir o subir a plataformas como **DockerLabs** [[21:39](#)].

Video original: [Cómo Crear LABORATORIOS VULNERABLES en DOCKER](#)