

1. Instalación de Docker en Kali Linux

Para una instalación limpia y fiable, se recomienda instalar Docker directamente desde los repositorios oficiales de Docker, en lugar de usar la versión predeterminada de Kali/Debian, ya que suele estar desactualizada.

Pasos de Instalación Recomendados

1. Actualizar el sistema e instalar dependencias:

```
sudo apt update
```

Instalar Docker (desde los repositorios de Kali):

```
sudo apt install -y docker.io
```

Aunque algunos prefieren el método del repositorio oficial de Docker por ser más reciente, docker.io funciona perfectamente para estos laboratorios.

Habilitar e iniciar el servicio Docker:

```
sudo systemctl enable docker --now
```

(--now ← Doble guión)

Recordatorio Importante (Post-instalación)

Si no quieres usar sudo antes de cada comando docker (lo cual es muy recomendable en Kali para la comodidad de uso), debes **añadir tu usuario al grupo docker** y luego **reiniciar tu sesión** o el terminal:

```
sudo usermod -aG docker $USER
```

Después de esto, el comando para cargar tu laboratorio, por ejemplo, **Psycho**, será tan simple como:

```
docker pull dockerlabs/psycho
docker run -d -p 80:80 dockerlabs/psycho
```

Si prefieres instalarlo desde Docker:

Añadir la clave GPG oficial de Docker:

```
curl -fsSL https://download.docker.com/linux/debian/gpg |  
sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-  
keyring.gpg
```

2. Configurar el repositorio estable de Docker:

```
echo "deb [arch=$(dpkg --print-architecture) signed-  
by=/usr/share/keyrings/docker-archive-keyring.gpg]  
https://download.docker.com/linux/debian ${lsb_release -  
cs} stable" | sudo tee /etc/apt/sources.list.d/docker.list  
> /dev/null
```

3. Instalar Docker Engine:

```
sudo apt update  
sudo apt install -y docker-ce docker-ce-cli containerd.io
```

4. Verificar la instalación:

```
sudo service docker start  
sudo docker run hello-world
```

Si ves el mensaje de bienvenida, la instalación es correcta.

5. Opcional (Recomendado): Usar Docker sin sudo

Para evitar tener que escribir sudo antes de cada comando docker, añade tu usuario al grupo docker. Cierra sesión y vuelve a iniciarla para que el cambio surta efecto.

```
sudo usermod -aG docker $USER
```

2. Cargar una Máquina de DockerLabs

Las "máquinas" de DockerLabs son imágenes Docker preconfiguradas para funcionar como laboratorios de seguridad o CTF. Una vez que Docker está instalado, el proceso para cargar y ejecutar una de estas máquinas es el siguiente:

Pasos para Cargar una Máquina

Los archivos `-auto_deploy.sh` e `injection.tar` nos dicen exactamente cómo poner en marcha el laboratorio "Injection". No necesitas usar el comando `docker pull` ya que ya tienes el archivo de imagen.

1. Archivos en la Descarga

1. `injection.tar`: Este es el archivo clave. Contiene la **imagen Docker** de la máquina **Injection** en un formato archivado (`.tar`). Es lo que la mayoría de los usuarios descargan manualmente cuando no quieren usar Docker Hub directamente.
2. `auto_deploy.sh`: Este es un **script de shell** (un archivo ejecutable) que el creador del laboratorio ha incluido para automatizar los pasos de carga y ejecución de la imagen. Es el método más sencillo y recomendado.

```
unzip injection.zip
```

2. Pasos para Desplegar la Máquina **Injection**

Debido a que tienes el script de despliegue, el proceso es muy rápido.

Paso A: Navegar a la Carpeta

Abre tu terminal de Kali y navega hasta la carpeta donde descomprimiste los archivos.

```
cd /ruta/a/la/carpeta/donde/estan/los/archivos
```

Paso B: Dar Permisos de Ejecución al Script

Por seguridad, los archivos de **shell** no suelen tener permisos de ejecución por defecto. Debes dárselos:

```
chmod +x auto_deploy.sh
```

Paso C: Ejecutar el Despliegue

Ejecuta el script. Este script se encargará automáticamente de dos cosas:

1. **Cargar** la imagen injection.tar en tu sistema Docker (el comando docker load).
2. **Ejecutar** la imagen como un contenedor, mapeando los puertos necesarios (el comando docker run).

```
sudo ./auto_deploy.sh <archivo.tar>
sudo ./auto_deploy.sh injection.tar
```

El script te mostrará mensajes de que está cargando la imagen y luego te confirmará el comando docker run que está utilizando.