

Puesta en marcha del entorno de prácticas

Curso 2024-2025. Versión: 1.0

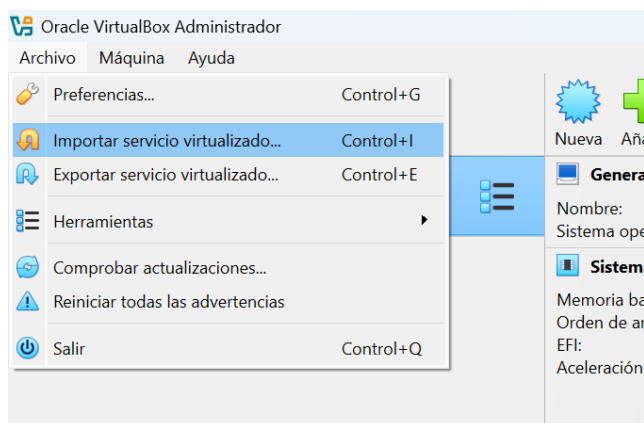
Importación de la máquina virtual

La práctica se realizará dentro de una máquina virtual que contiene todos los programas necesarios para su ejecución. Esta máquina virtual está configurada con un sistema operativo Ubuntu Desktop 24.04.2 LTS 64 bits con una memoria RAM de 4GBytes, un disco duro de 25GBytes y la paravirtualización KVM activada. Dentro de esta máquina virtual está instalado Docker y un conjunto de contenedores docker interconectados mediante redes virtuales dónde el/la estudiante tiene que realizar las acciones de la práctica.

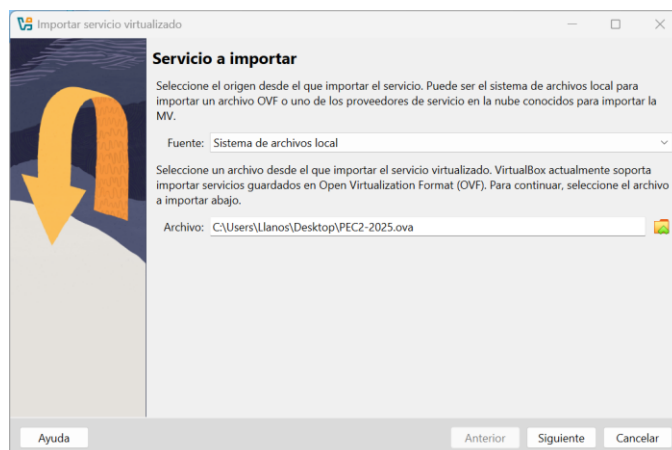
Para realizar la práctica el/la estudiante debe haber instalado en su equipo el programa de virtualización Virtual Box. Este programa se puede descargar de manera gratuita en el siguiente enlace: <https://www.virtualbox.org/wiki/Downloads>. Existen versiones disponibles para entornos Windows, OS X y Linux. Tras la descarga, la instalación es sencilla con ayuda del asistente.

Tras la instalación en el equipo de Virtual Box, debe descargar el fichero PEC2-2025.ova del curso virtual en el enlace habilitado para ello. Esta descarga es de aproximadamente 6,64 GBytes.

Arrancamos el programa Virtual Box. Una vez iniciado hacemos clic en “**Archivo**” en la parte superior derecha del programa y seleccionamos la opción “**Importar servicio virtualizado**” del menú que se despliega.



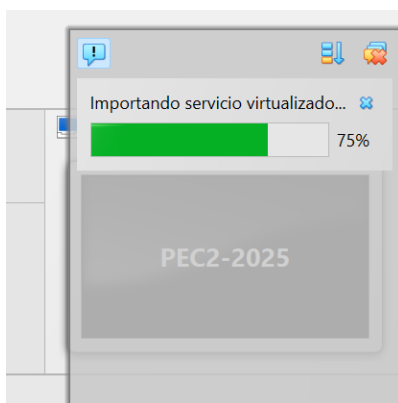
Esto inicia el asistente de importación. En la primera ventana debemos indicar que la *Fuente* es el *Sistema de archivos local*. Y en el campo de *Archivo* debemos localizar el fichero *PEC-2025.ova* en la carpeta dónde lo descargamos. Para ello podemos abrir un diálogo de ficheros haciendo clic en la carpeta con la flecha verde.



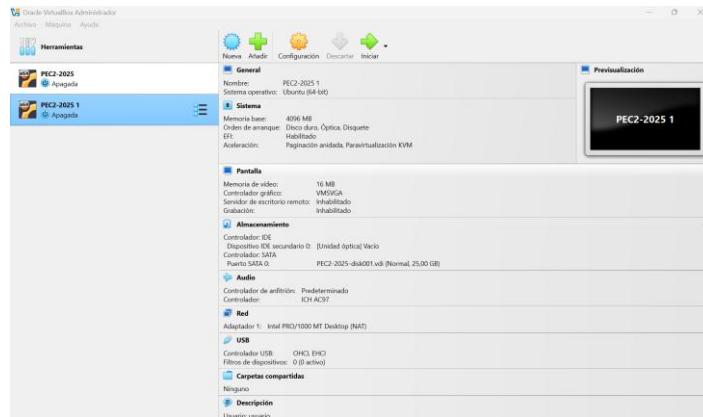
Una vez completado para continuar hacemos clic en el botón *Siguiente*. El siguiente cuadro de diálogo configura la máquina virtual a importar. En principio debemos dejar las opciones por defecto. Si queremos podemos cambiar la carpeta dónde va a residir la máquina virtual en la opción *Carpeta base de máquina*. Pero el resto de los parámetros deben quedarse como están.



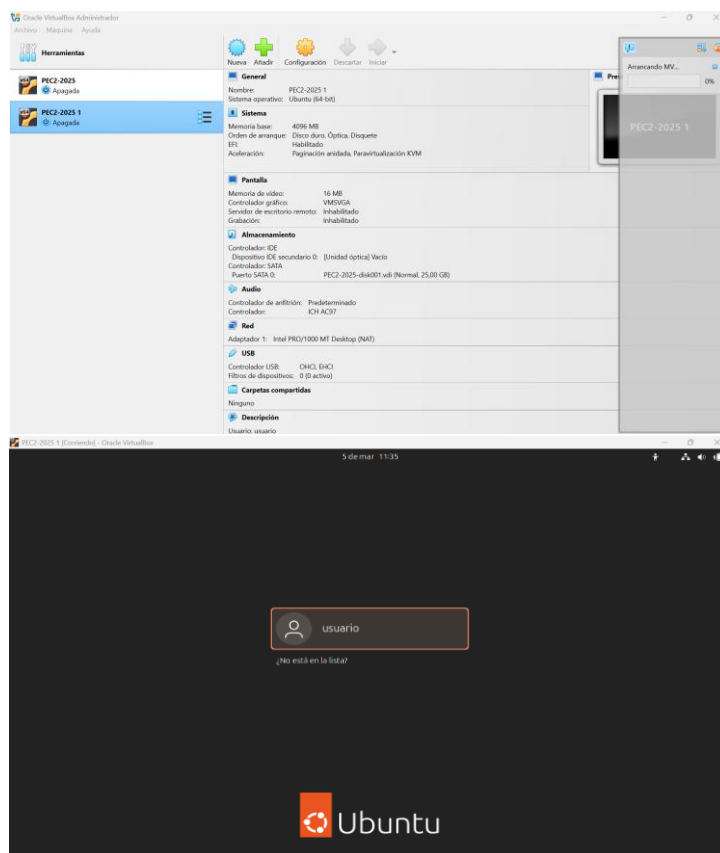
Continuamos haciendo clic en el botón de *Terminar*. Comienza la importación. En la parte inferior del diálogo se nos indica el tiempo estimado de duración, que dependerá de la configuración de nuestro equipo.



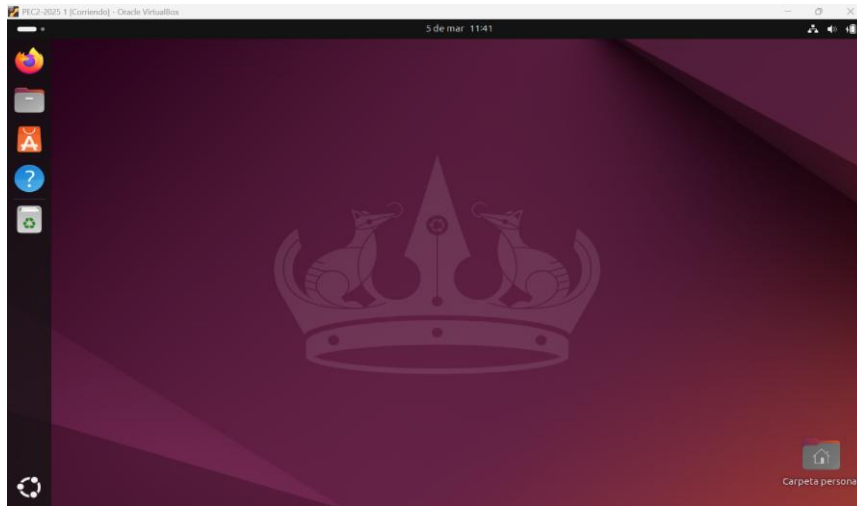
Una vez finalizado el proceso de importación encontraremos la máquina en el listado de máquinas virtuales disponibles, en la parte derecha de la ventana principal de Virtual Box.



Ya podemos iniciar la máquina virtual haciendo doble clic en el nombre de la máquina virtual. Se abrirá una nueva ventana dónde arrancará la máquina virtual:




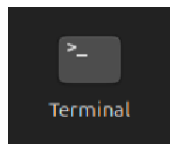
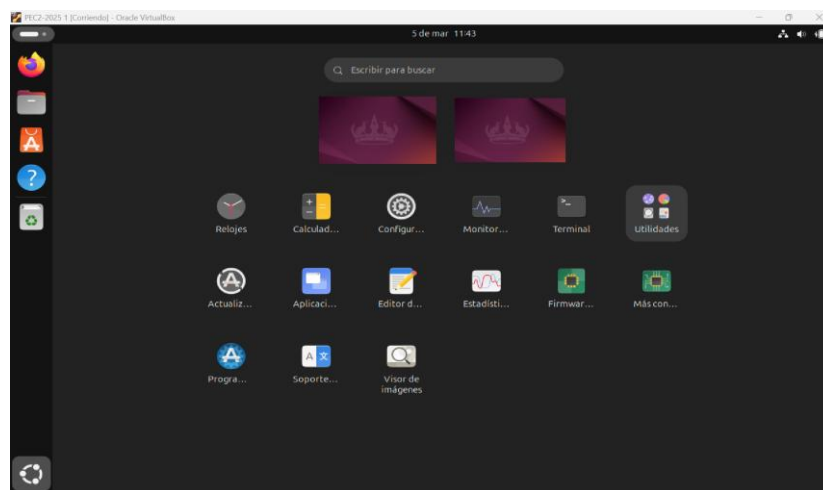
El usuario es usuario y la contraseña es \$egurid@d. También puedes encontrar estos datos en los comentarios de la máquina virtual.

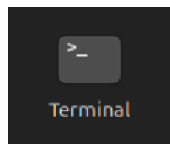


A la hora de realizar la práctica puede utilizar o bien una interfaz web (experimental) o bien el uso del entorno mediante comandos. Elige sólo una de estas opciones.

Uso de la interfaz web (experimental)

Lo primero debemos abrir un Terminal desde Ubuntu. Hacemos clic en el botón  en la parte inferior izquierda del escritorio. Nos abrirá una ventana dónde se nos mostrarán las aplicaciones más frecuentes.



Entre ellas se encuentra Terminal . La seleccionamos y se nos abrirá un terminal de trabajo.

Desde este terminal haremos usaremos las siguientes ordenes:

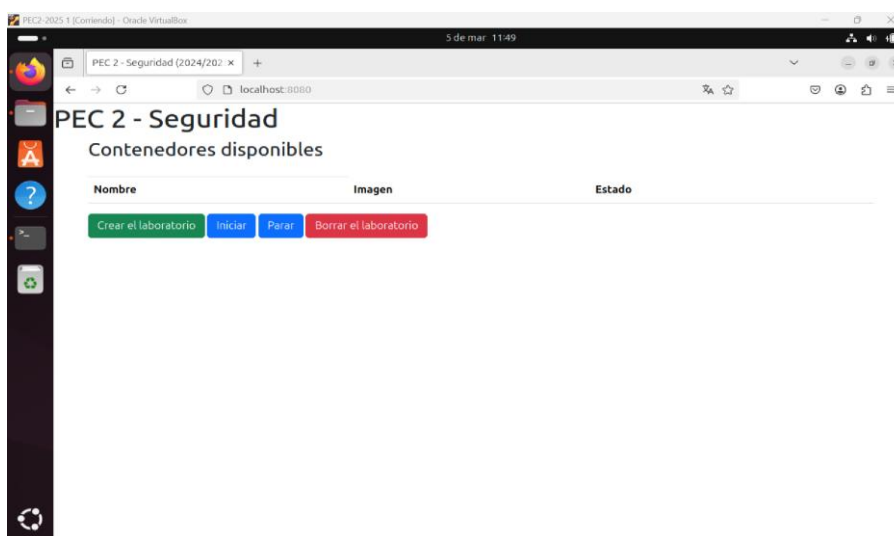
cd lab

sudo python3 weblab.py

para arrancar el entorno web de prácticas. Nos pedirá la misma contraseña que usamos para iniciar sesión en la máquina virtual.

```
usuario@PEC2-2025:~$ cd lab
usuario@PEC2-2025:~/lab$ sudo python3 weblab.py
WebSocket transport not available. Install simple-websocket for improved performance.
* Serving Flask app '__name__'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Running on http://127.0.0.1:8080
Press CTRL+C to quit
* Restarting with stat
WebSocket transport not available. Install simple-websocket for improved performance.
* Debugger is active!
* Debugger PIN: 120-225-493
```

Una vez puesto en marcha podemos abrir el navegador Firefox disponible en la barra de iconos arriba a la izquierda del escritorio y acceder a la página web <http://localhost:8080>.



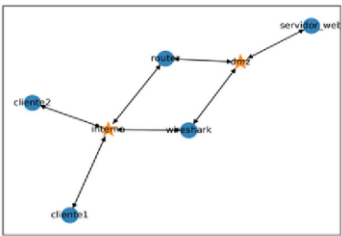
El laboratorio cuenta con cuatro botones básicos:

- **Crear el laboratorio:** crea los contenedores y los inicia. Es el botón que debemos usar para iniciar las prácticas. Dado que inicia el entorno de prácticas y lo pone en ejecución es posible que tarde algunos instantes y debamos recargar la página. Tras su ejecución la apariencia de la página es la siguiente:

PEC 2 - Seguridad (2024/2025) +

localhost:8080

PEC 2 - Seguridad



Contenedores disponibles

Nombre	Imagen	Estado
cliente2	alpine:latest	running
servidor_web	seigetsu/httpd-sshd:2.4-alpine	running
cliente1	alpine:latest	running
wireshark	lscr.io/linuxserver/wireshark:latest	running
router	alpine:latest	running

[Consola](#) [Consola](#) [Consola](#) [Wireshark](#) [Consola](#)

[Crear el laboratorio](#) [Iniciar](#) [Parar](#) [Borrar el laboratorio](#)

Se muestra un esquema de la red de la práctica y listado con los contenedores de la práctica, su estado y un enlace para acceder a la consola o la herramienta Wireshark.

- **Parar:** nos permite detener el laboratorio, guardando todos los avances realizados en la práctica hasta ese momento. Igual que ocurría con otras funciones de la página web podemos encontrarnos con que tarde un tiempo en detener el laboratorio y sea necesario recargar la página web. Tras su ejecución el aspecto de la página web es el siguiente:

PEC 2 - Seguridad (2024/2025) +

localhost:8080

PEC 2 - Seguridad

Contenedores disponibles

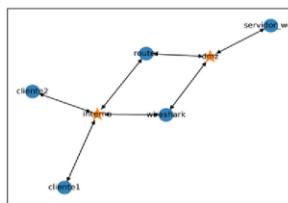
Nombre	Imagen	Estado
cliente2	alpine:latest	exited
servidor_web	seigetsu/httpd-sshd:2.4-alpine	exited
cliente1	alpine:latest	exited
wireshark	lscr.io/linuxserver/wireshark:latest	exited
router	alpine:latest	exited

[Crear el laboratorio](#) [Iniciar](#) [Parar](#) [Borrar el laboratorio](#)

Hay que destacar que se listan los contenedores, pero su estado es “exited” y no disponemos del enlace de acceso a la consola.

- **Iniciar:** permite arrancar de nuevo el entorno de prácticas que se ha detenido con el botón de parar. Igual que ocurría con otras funciones de la página web podemos encontrarnos con que tarde un tiempo en detener el laboratorio y sea necesario recargar la página web. Tras su ejecución el aspecto de la página web es el siguiente:

PEC 2 - Seguridad



Contenedores disponibles

Nombre	Imagen	Estado	
cliente2	alpine:latest	running	Consola
servidor_web	seigetsu/httpd-sshd:2.4-alpine	running	Consola
cliente1	alpine:latest	running	Consola
wireshark	lscr.io/linuxserver/wireshark:latest	running	Wireshark
router	alpine:latest	running	Consola

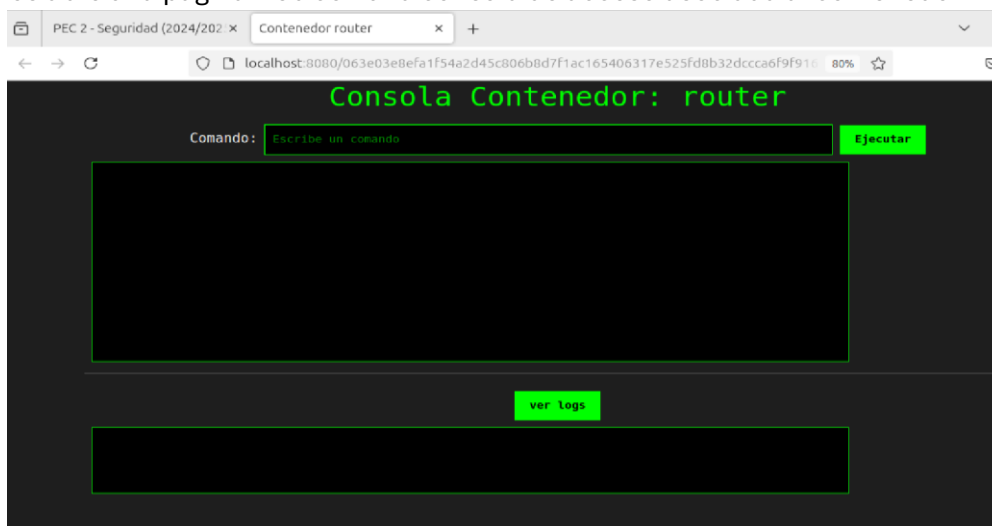
[Crear el laboratorio](#)
[Iniciar](#)
[Parar](#)
[Borrar el laboratorio](#)

Volvemos a ver que los contenedores están en el estado ‘running’ y aparecen los enlaces de acceso a la consola de cada contenedor.

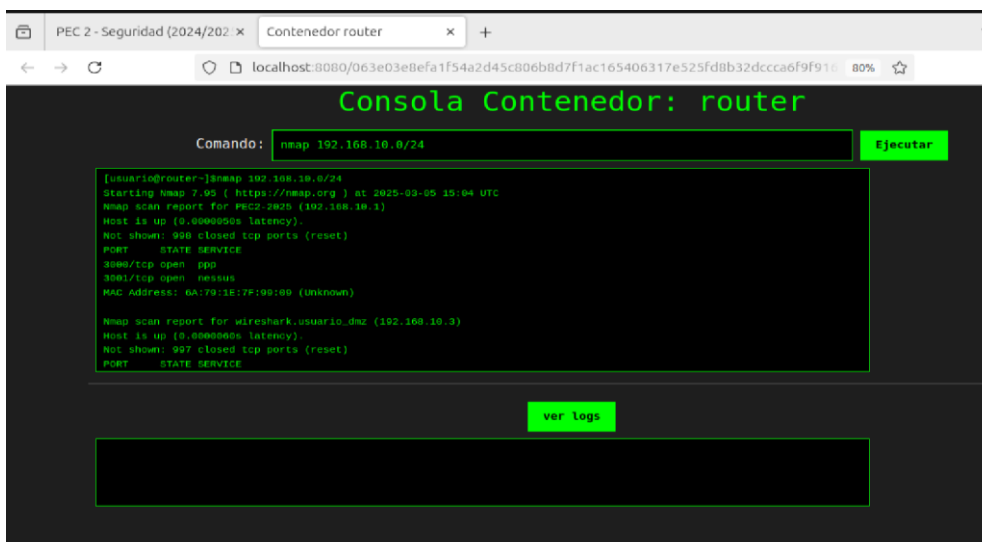
- **Borrar el laboratorio:** detiene los contenedores de la práctica y los borra por completo. **Es una acción irreversible y se perderá todo el avance de la práctica.** Se recomienda usar esta opción o bien porque hemos terminado completamente la práctica o bien porque deseamos volver a comenzar la práctica desde el principio. Igual que ocurriría con otras funciones de la página web podemos encontrarnos con que tarde un tiempo en detener el laboratorio y sea necesario recargar la página web. El aspecto de la página web tras ejecutar este comando es el siguiente:



Una vez arrancado el entorno de prácticas disponemos para todos los contenedores de la práctica a excepción del contenedor que contiene Wireshark un enlace llamado Consola que nos abre una página web con una consola de acceso asociada al contenedor:

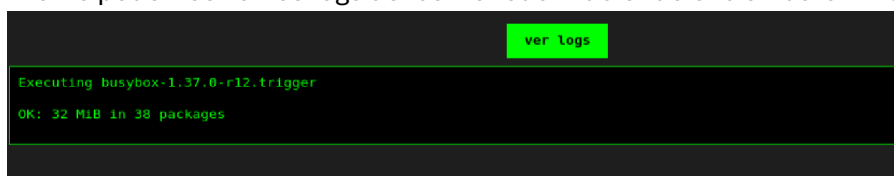


En el cuadro “Comando” podemos escribir la orden que queremos ejecutar en dicho contenedor y se ejecuta al hacer clic en el botón “Ejecutar”. El resultado del comando lo podremos ver en el cuadro bajo este formulario.



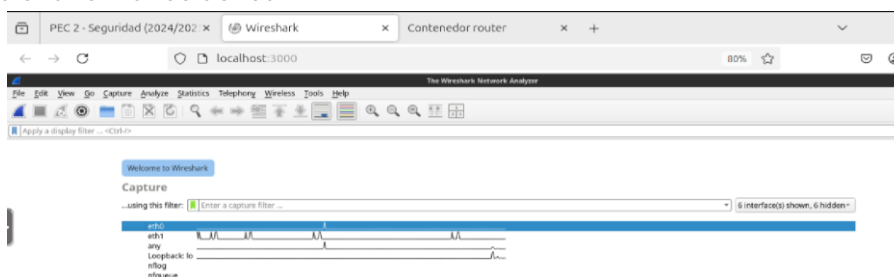
Dado que la comunicación con el contenedor para ejecutar los comandos no se realiza mediante ssh u otro servicio para evitar tener puertos abiertos en los contenedores que se bloqueen cuando se configuren los cortafuegos es posible que el resultado de la ejecución de un comando tarde unos instantes así que rogamos seáis pacientes cuando ejecutéis los comandos.

Adicionalmente podemos ver los logs del contenedor haciendo clic en botón “ver logs”.

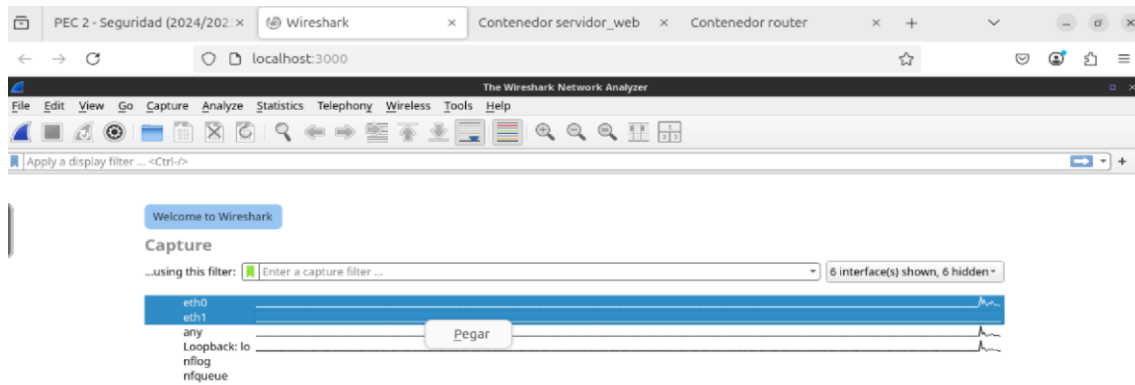


Podemos tener tantas pestañas con tantas consolas cómo contenedores tenemos en el laboratorio.

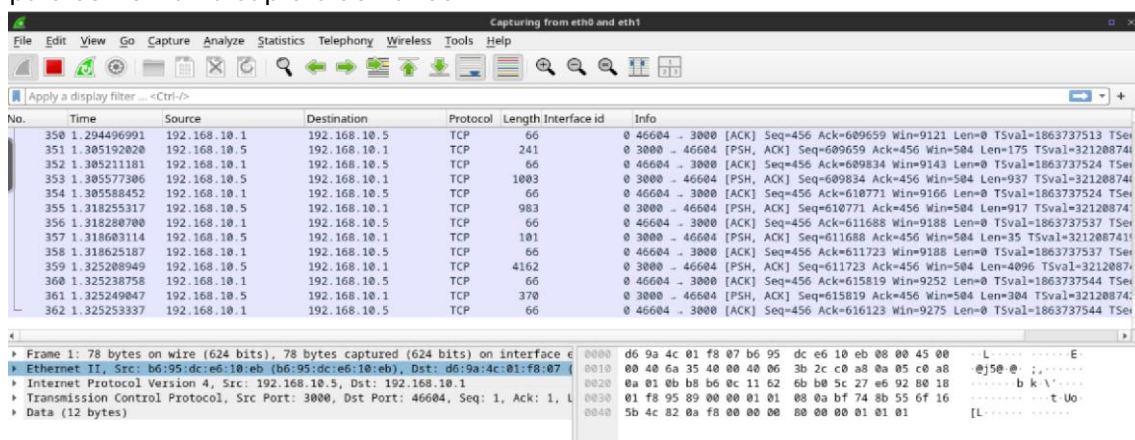
Un contenedor especial es Wireshark. Nos permite tener en una pestaña una instancia de la herramienta Wireshark conectada a las dos redes existentes en el entorno de prácticas e inspeccionar el tráfico de red.



Al abrir su pestaña se nos muestran las interfaces dónde podemos capturar el tráfico. Es importante seleccionar tanto eth0 como eth1 para poder ver el tráfico de ambas redes.




Una vez seleccionadas ambas redes podemos hacer clic en el icono de la aleta azul para comenzar la captura de tráfico.

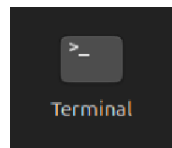
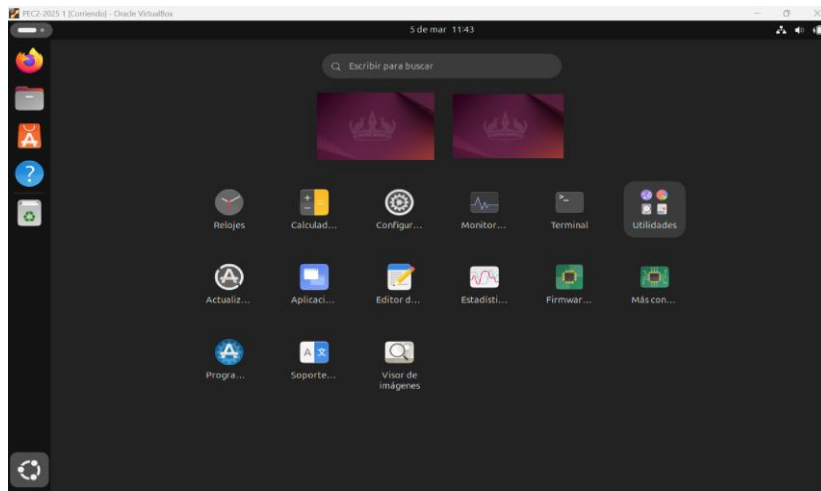


Contamos con una columna llamada Interface id para poder identificar en qué interfaz de red se produce la comunicación. En cualquier momento podemos detener la captura de información con el icono cuadrado rojo y analizar el tráfico capturado.

Uso del entorno de comandos Docker

Alternativamente a la opción anterior más experimental, y si lo preferimos, podemos trabajar directamente desde un terminal mediante comandos. Lo que no recomendamos es utilizar ambas opciones a la vez.

Lo primero debemos abrir un Terminal desde Ubuntu. Hacemos clic en el botón  en la parte inferior izquierda del escritorio. Nos abrirá una ventana dónde se nos mostrarán las aplicaciones más frecuentes.



Entre ellas se encuentra Terminal. La seleccionamos y se nos abrirá un terminal de trabajo.

Una vez en el terminal para arrancar por primera vez el entorno de prácticas debemos usar el comando **sudo docker compose up -d**.

```
usuario@PEC2-2025:~$ sudo docker compose up -d
[+] Running 5/7
 ✓ Network usuario_interna Created 0.4s
 ✓ Network usuario_dmz Created 0.8s
  Container wireshark Starting 11.7s
  Container router Starting 11.7s
 ✓ Container servidor_web Created 1.8s
 ✓ Container cliente1 Created 1.8s
 ✓ Container cliente2 Created 1.8s

usuario@PEC2-2025:~$ sudo docker compose up -d
[+] Running 7/7
 ✓ Network usuario_interna Created 0.4s
 ✓ Network usuario_dmz Created 0.8s
 ✓ Container wireshark Started 47.8s
 ✓ Container router Started 12.0s
 ✓ Container servidor_web Started 40.6s
 ✓ Container cliente1 Started 36.8s
 ✓ Container cliente2 Started 40.0s
usuario@PEC2-2025:~$
```

La opción **up** crea los contenedores y los inicia. La opción **-d** es para ejecutarlo en segundo plano y poder disponer de la consola para poder ejecutar otros comandos.

Una vez iniciado el entorno de prácticas podemos comprobar que todos los contenedores están en ejecución correctamente con la orden **sudo docker ps -a**.

```
usuario@PEC2-2025:~$ sudo docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	NAMES	CREATED	STATUS
dc9609ddd4b	alpine	"sh -c 'apk add --no..."	cliente1	About a minute ago	Up About a minu
0452c4676057	seigetsu/httpd-sshd:2.4-alpine	"httpd-sshd-entrypoin..."	servidor_web	About a minute ago	Up About a minu
767207e0722f	alpine	"sh -c 'apk add --no..."	cliente2	About a minute ago	Up About a minu
37fd3ccd7515	alpine	"sh -c 'apk add --no..."	router	About a minute ago	Up About a minu
6980ba610594	lscr.io/linuxserver/wireshark:latest	"/init"	wireshark	About a minute ago	Up About a minu

En el caso de este entorno de prácticas debéis tener disponibles cinco contenedores: servidor_web, router, wireshark, cliente1 y cliente2.

A partir de este momento puede trabajar con el entorno de prácticas. Si abrimos un navegador en la máquina virtual podemos acceder a Wireshark usando la url <http://localhost:3000>.

Si queremos comprobar los logs de algún contenedor lo haremos con la orden **docker logs <nombre_contenedor>** donde nombre_contenedor es el nombre del contenedor. Por ejemplo, si queremos ver los logs del servidor web, cuyo contenedor es servidor_web, usaremos la orden *docker logs servidor_web*:

```
usuario@PEC2-2025:~$ sudo docker logs servidor_web
/usr/local/bin/httpd-sshd-entrypoin.sh start.
chpasswd: password for 'root' changed
Generating public/private rsa key pair.
Your identification has been saved in /etc/ssh/ssh_host_rsa_key
Your public key has been saved in /etc/ssh/ssh_host_rsa_key.pub
The key fingerprint is:
SHA256:tmo8+eHPaKEynVoN/GV7z7PA2REp1ES6QjX0oOG4i5g root@0452c4676057
The key's randomart image is:
+---[RSA 3072]-----+
|      . . ++o      |
|      o o * o..    |
|      . o . = o     |
|      . . . o .    |
|      + S + . .    |
|      o . *.+ + o . |
|      E .oo+=.. = . |
|      o.Oo.+ . . +. |
|      .+.++o  +o    |
+-----[SHA256]-----+
Generating public/private dsa key pair.
Your identification has been saved in /etc/ssh/ssh_host_dsa_key
Your public key has been saved in /etc/ssh/ssh_host_dsa_key.pub
The key fingerprint is:
```

Para poder acceder a la consola de cada contenedor debemos usar la orden **docker exec --privileged -it <nombre_contenedor> <terminal>**. La opción privileged nos da acceso con permisos de root en la consola. La opción it nos permite abrir un nuevo terminal interactivo. El nombre_contenedor será el nombre del contenedor al que quiero conectarme. Y terminal se asocia a la consola correspondiente. En nuestra red la consola de los contenedores es sh.

```
usuario@PEC2-2025:~$ sudo docker exec --privileged -it router sh
/ # ls
bin    etc    lib    mnt    proc   run    srv    tmp    var
dev    home   media  opt    root   sbin   sys    usr
/ # exit
usuario@PEC2-2025:~$
```

Para finalizar esta sesión se usa el comando exit.

Para finalizar una sesión, pero retomarla más tarde, usaremos el comando **sudo docker-compose stop**.

```

usuario@PEC2-2025:~$ sudo docker compose stop
[+] Stopping 1/4
  Container cliente1      Stopping
  Container cliente2      Stopping
  Container servidor_web  Stopped
  Container wireshark     Stopping

usuario@PEC2-2025:~$ sudo docker compose stop
[+] Stopping 5/5
  Container cliente1      Stopped
  Container cliente2      Stopped
  Container servidor_web  Stopped
  Container wireshark     Stopped
  Container router        Stopped
usuario@PEC2-2025:~$

```

De esta manera se detienen de manera ordenada los contenedores almacenando el avance conseguido en la práctica. Para retomar la práctica con los mismos contenedores usaremos la orden **sudo docker compose start**.

```

usuario@PEC2-2025:~$ sudo docker compose start
[+] Running 1/5
  Container wireshark     Starting
  Container router        Started
  Container cliente1      Starting
  Container cliente2      Starting
  Container servidor_web  Starting

usuario@PEC2-2025:~$ sudo docker compose start
[+] Running 5/5
  Container wireshark     Started
  Container router        Started
  Container cliente1      Started
  Container cliente2      Started
  Container servidor_web  Started
usuario@PEC2-2025:~$

```

Por último, si queremos reiniciar completamente el entorno o ya hemos terminado con la práctica y queremos borrar los contenedores usaremos **sudo docker compose down**:

¡OJO! Esta opción borra todos los contenedores y todos los avances realizados.

```

usuario@PEC2-2025:~$ sudo docker compose down
[+] Running 0/4
  Container servidor_web  Stopping      5.8s
  Container wireshark     Stopping      5.8s
  Container cliente1      Stopping      5.8s
  Container cliente2      Stopping      5.8s

usuario@PEC2-2025:~$ sudo docker compose down
[+] Running 7/7
  Container servidor_web  Removed      18.7s
  Container wireshark     Removed      33.5s
  Container cliente1      Removed      26.9s
  Container cliente2      Removed      27.0s
  Container router        Removed      15.3s
  Network usuario_dnz     Removed      6.0s
  Network usuario_interna Removed      4.0s

```

Resumen de los comandos:

Comando	Uso
sudo docker compose up -d	Crear los contenedores desde cero e iniciar los contenedores

sudo docker ps -a	Ver los contenedores y su estado
sudo docker logs <nombre_contenedor>	Ver los logs de ejecución del contenedor
sudo docker exec --privileged -it <nombre> <consola>	Acceder a la consola de un contenedor para ejecutar instrucciones en su interior. Se sale de esta sesión con exit.
sudo docker compose stop	Detener la ejecución de los contenedores, sin borrar los avances.
sudo docker compose start	Reiniciar el entorno de prácticas que se ha detenido anteriormente con la orden docker compose stop.
sudo docker compose down	Detener el entorno de prácticas y borrar los contenedores. Se borran los contenedores y todos los avances. Tras este comando es necesario usar sudo docker compose up -d para volver a poner en marcha el entorno de prácticas.