


Práctica I: Tecnologías de Virtualización	 Universidad Carlos III de Madrid
Redes Software Última modificación : 05-02-2025	2024-25

Introducción y objetivos

1. Objetivos

El objetivo de esta práctica es familiarizarse con los entornos de virtualización y entornos de contenedores en el sistema operativo Linux, y entender las diferencias entre ambos.

Para ello, se empieza por crear la Máquina Virtual (VM) que se va a utilizar en las restantes prácticas de la asignatura. En la segunda parte de la práctica, se utiliza una copia de dicha VM para experimentar con entornos de contenedores en el Sistema Operativo (SO) Linux.

2. Normas básicas

Estudie este documento en profundidad antes de empezar el laboratorio. Asegúrese que lo entiende en su totalidad. El laboratorio está pensado para realizarse en grupos de dos estudiantes. Aclare posibles dudas con los profesores en el laboratorio.

Hito XX

Los **hitos** aparecen resaltados de esta manera. En el texto del hito se indican los resultados que se tienen que incluir en el entregable.

3. Entorno de trabajo

Es necesario que prepare el equipo para realizar la práctica *antes* de la sesión. Tenga en cuenta que tanto el programa hipervisor como la imagen de instalación del SO están empaquetados en ficheros de tamaño considerable y que, por tanto, **es probable que sea inviable descargarlos durante la sesión de laboratorio**.

Tanto el hipervisor como la imagen de SO están condicionados por su equipo. Se va a trabajar con una imagen reciente de la distribución Ubuntu versión 22.04 (código de distribución jammy) para servidores.

3.1. Equipos con procesador (CPU) Intel

En equipos con CPU compatible Intel (amd64), se recomienda instalar el programa VirtualBox [1] en su equipo. Otras alternativas son:

1. En equipos con el SO Windows 11, existe la posibilidad de utilizar el hipervisor Hyper-V[2] en lugar de VirtualBox, siempre que se pueda activar en la configuración del equipo.
2. En equipos Apple con procesador Intel, se puede utilizar el programa UTM [3], al igual que en los equipos más modernos de dicho fabricante, que utilizan procesadores compatibles con ARM.

Esta se puede obtener a través del enlace <https://releases.ubuntu.com/jammy/ubuntu-22.04.5-live-server-amd64.iso>

3.2. Equipos modernos de Apple

En el caso de que su equipo sea del fabricante Apple y esté equipado con CPU M1, M2 o M3, se deberá utilizar el programa UTM [3] y la imagen <https://cdimage.ubuntu.com/releases/jammy/release/ubuntu-22.04.5-live-server-arm64.iso>

En el caso de un MacBook con CPU de Intel, también es recomendable utilizar el programa hipervisor UTM. En estos equipos, se utilizará la imagen indicada en la sección 3.1.

3.3. Después de instalar la VM

Una vez haya creado la imagen inicial, la operación de la VM es la misma en todos los casos. Está basada en el entorno gráfico Mate como entorno de ventanas por su reducido consumo de recursos. Además del software necesario para los laboratorios, la máquina que va a crear incluye un número suficiente de programas auxiliares para ayudar en la ejecución de este y los demás laboratorios. **No instale más paquetes a no ser que se lo indiquen los profesores de laboratorio.**

4. Estructura del documento

Este documento está estructurado de la siguiente manera:

1. En el capítulo 1 se introduce el uso del hipervisor VirtualBox (o UTM, si estuviera utilizando un equipo de Apple) y se crea la VM que se utilizará en el resto de los laboratorios.
2. En el capítulo 2 se ve cómo funcionan los contenedores.
3. En el apéndice A se dan consejos para activar las extensiones de VirtualBox en la VM.

1. Creación de la VM para los laboratorios de la asignatura

La componente práctica de Redes Software se realiza sobre el propio equipo del alumno. Utilizamos una VM donde se ejecuta el sistema operativo Linux y en la que se instalan un controlador OpenFlow y un emulador de redes. En esta primera práctica, nos vamos a familiarizar con los conceptos básicos de la virtualización creando dicha VM.

El proceso se divide en dos partes:

1. Creación de la VM
2. Instalación del entorno de desarrollo para las prácticas Software Defined Networking (SDN)

1.1. Creación de la VM

Esta parte de la práctica depende del equipo sobre el que se vaya a realizar. El objetivo de la misma es obtener una imagen inicial, familiarizándose con los conceptos básicos de la virtualización. Como ya se ha explicado en la introducción a la práctica, **es necesario que haya descargado los ficheros necesarios antes de empezar.**

1.1.1. Creación de la VM con el hipervisor VirtualBox

En cualquier equipo con procesador compatible Intel, independientemente del Sistema Operativo, se recomienda utilizar el programa VirtualBox como hipervisor de tipo 2. Para poder realizar la práctica, se tendrá que tener instalado dicho programa. Las capturas de pantalla que se incluyen para guiar en el proceso se han obtenido con la **versión 7.1** del programa en un equipo con el SO Linux, distribución Ubuntu 22.04.

El primer paso es abrir el programa VirtualBox y seleccionar 'Crear una VM nueva' en el diálogo inicial:

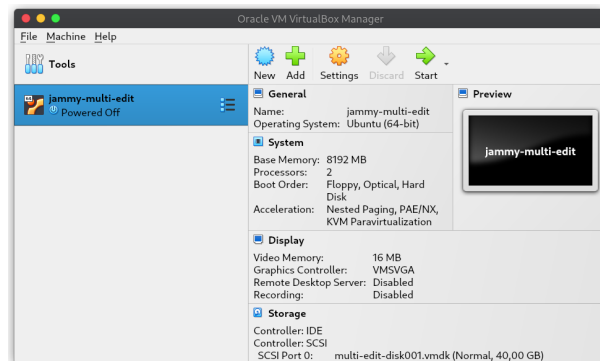
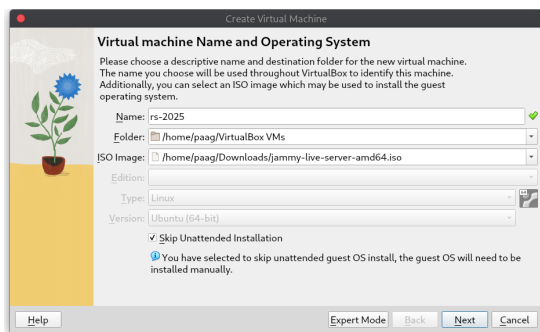


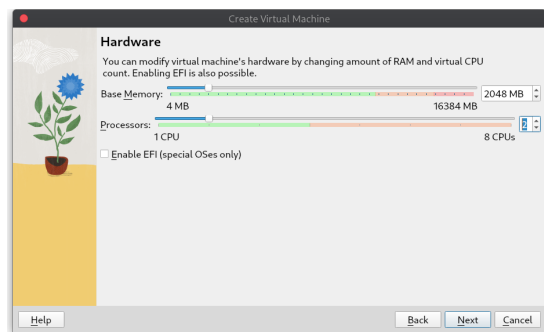
Figura 1.1.: Diálogo inicial del programa

Se abrirá el diálogo de creación de VMs, en el que le pondremos a la VM el nombre `rs-2025` y seleccionaremos la imagen ISO del SO Ubuntu 22.04 server que nos hemos bajado previamente.

Como se puede apreciar, VirtualBox detecta directamente el tipo de Sistema Operativo. También se activará ignorar la instalación automática. A continuación se seleccionará pasar a la siguiente pantalla, que nos pedirá la memoria y el número de CPUs virtuales. Se recomienda subir el número de CPUs a 2:



(a) Creación de una VM

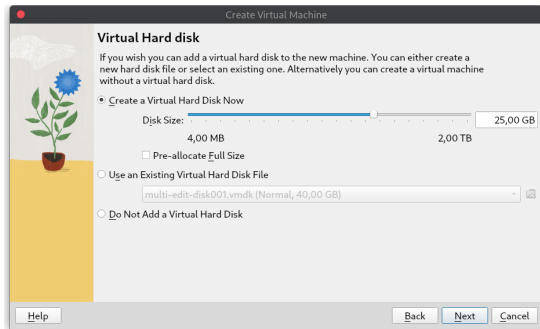


(b) Selección de memoria y CPUs virtuales

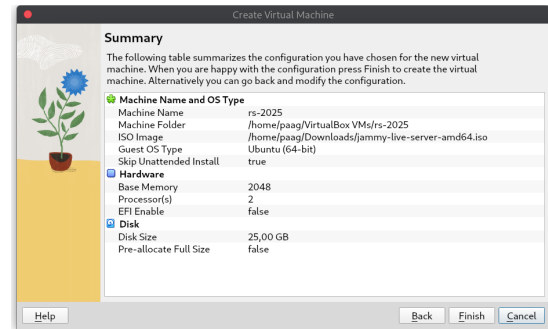
El siguiente paso es seleccionar el tamaño del disco duro virtual sobre el que se instalará el SO.

En función de los recursos disponibles en su equipo, se puede reducir ligeramente el tamaño propuesto, **pero nunca por debajo de 20 GBytes**.

Llegados a este punto, VirtualBox ofrecerá un diálogo de resumen. Una vez comprobado que los datos coinciden con lo que hemos seleccionado, se puede **finalizar** el proceso:



(a) Selección del tamaño del disco duro

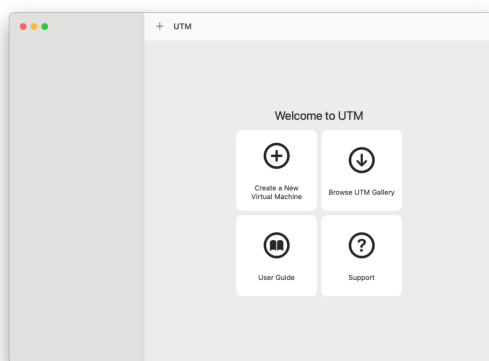


(b) Diálogo con el resumen de características

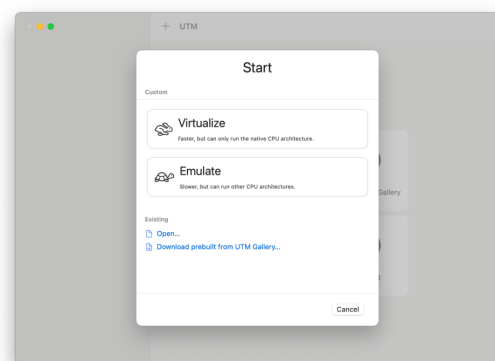
1.1.2. Creación de la VM con el hipervisor UTM

En equipos Apple con hardware de Apple, se tendrá que utilizar otro programa hipervisor, dado que VirtualBox está todavía en proceso de desarrollo¹. Por ello, en este caso, el programa utilizado será UTM[3] como hipervisor de tipo 2. Las capturas se han obtenido sobre un equipo con procesador Intel, la versión *Monterrey* del SO macOS y la **versión 4.5.3** del programa.

Una vez se arranque el programa, se seleccionará crear una VM nueva y virtualizar (en vez de emular; este último modo permite crear una VM para procesadores distintos al instalado en el sistema):



(a) Pantalla principal de UTM

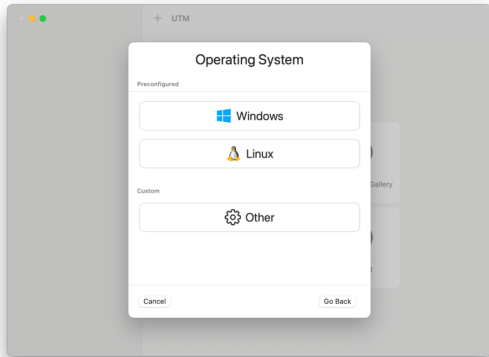


(b) Seleccionar el modo de virtualización

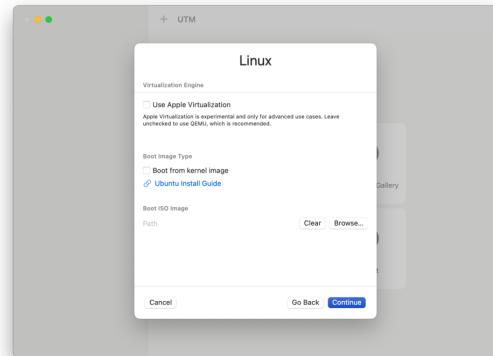
¹ Hay una versión *beta* disponible, pero no se puede garantizar un funcionamiento suficientemente estable.

Tecnologías de Virtualización

En este punto, indicaremos que queremos crear una VM con un SO Linux a partir de la imagen que nos acabamos de descargar:

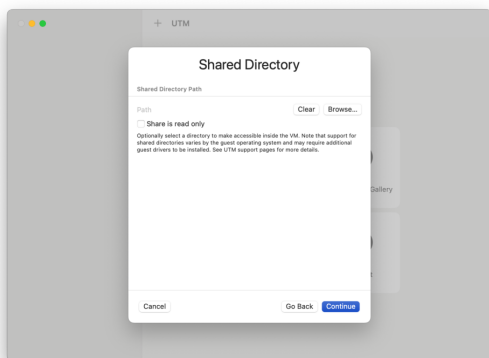


(a) Seleccionar Linux como SO huésped

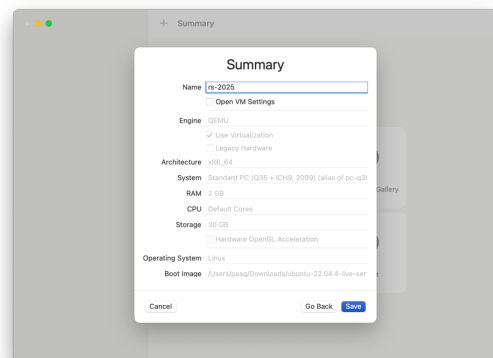


(b) Seleccionar la imagen ISO descargada previamente

Por ahora no se configurará ningún directorio compartido y se cambiará el nombre de la VM a rs-2025



(a) No configurar ningún directorio compartido



(b) Cambiar el nombre a la VM

La VM estará lista para arrancar:

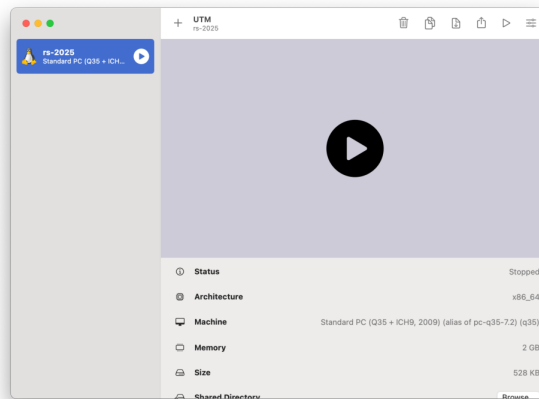


Figura 1.7.: La VM preparada para el arranque

En este momento, la VM estará lista para arrancar e instalar el Sistema Operativo.

1.1.3. Instalación del SO básico

En este punto, la instalación del SO Ubuntu server es la misma para VirtualBox y UTM. De hecho, el proceso es el que se sigue para instalarlo en un equipo real, partiendo de un medio de instalación físico como podría ser un soporte óptico (e.d. un DVD) o un disco externo (un «pincho» USB) en el que hayamos grabado la imagen que descargamos de Internet.

Una vez arrancada la VM que acaba de crear, aparecerá el menú de arranque de la imagen, en el que seleccionaremos “Try or install Ubuntu Server”:

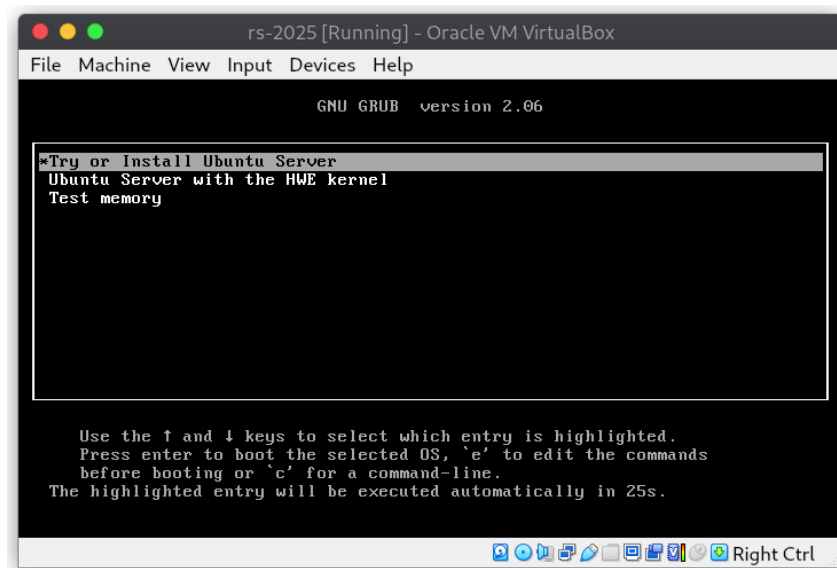
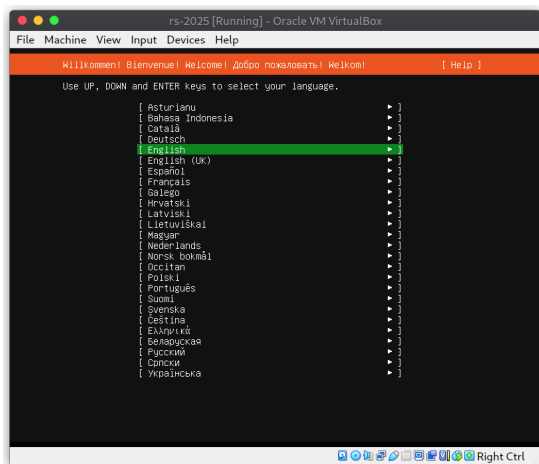
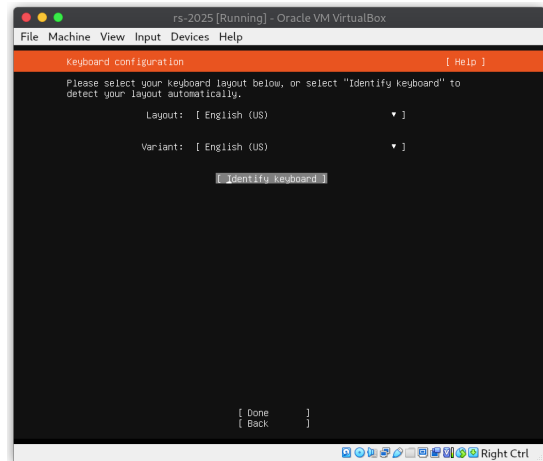


Figura 1.8.: Arranque inicial

Después del proceso de arranque, aparecerá la pantalla de selección del idioma de instalación. Se recomienda mantener la selección por defecto (**English**). Pulsar la tecla «Enter» para seguir. Se pasará al diálogo de selección del teclado. Utilizando la tecla de tabulación, seleccionar la opción de identificación del teclado y seguir el diálogo hasta que se haya detectado correctamente. Como se podrá apreciar, cada vez que se pulsa el tabulador, se resalta un campo distinto:



(a) Selección del idioma



(b) Seleccionando el diálogo de identificación del teclado.

Una vez se termine el proceso, utilizando las teclas de cursor o el tabulador, seleccionar «Done».

En el siguiente diálogo, seleccionar «Ubuntu Server» como base para la instalación y a continuación, al finalizar cada uno de los siguientes paso, seleccionar «Done»:

1. comprobar que se detecta correctamente la interfaz de **virtual**
2. dejar vacío el campo de dirección del Proxy (puesto que no se utiliza ninguno)
3. dejar que el instalador compruebe el repositorio de Ubuntu que selecciona por defecto
4. seleccionar la configuración de disco por defecto
5. confirmar la configuración el instalador ha detectado
6. confirmar que se quiere realizar la acción seleccionando el campo «Continue»:

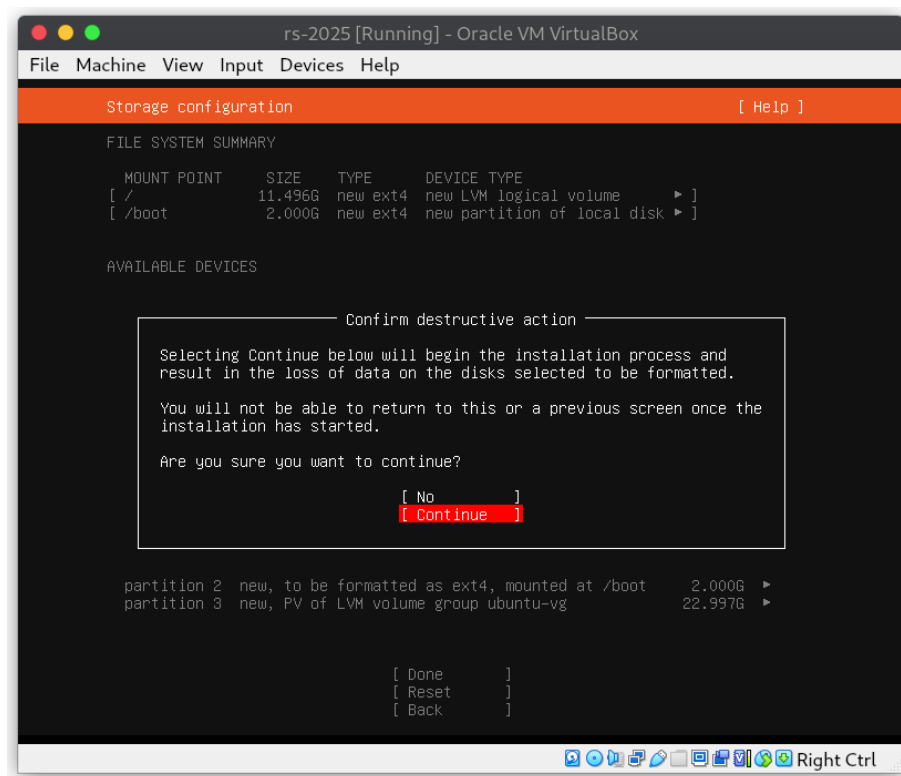


Figura 1.10.: Seleccionar la configuración del disco virtual

7. en el diálogo de definición del perfil del usuario:
 - a) Poner Student para el nombre del usuario
 - b) Poner rs-2025 para el nombre del servidor
 - c) Poner student para el identificador de usuario (*username*)
 - d) Poner y confirmar 5tud3nt como clave para el usuario

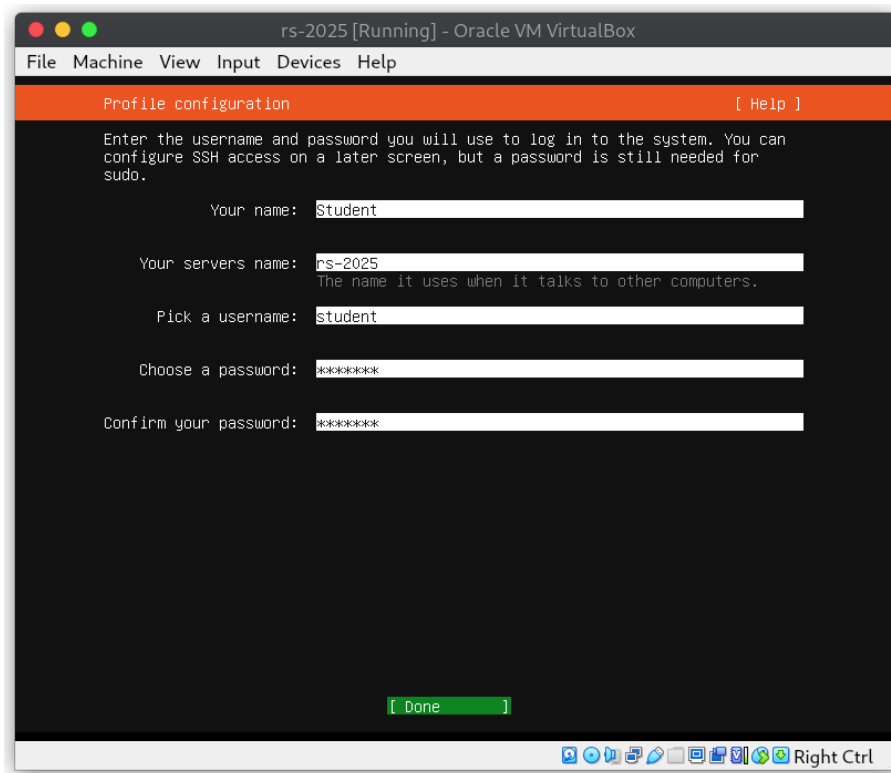
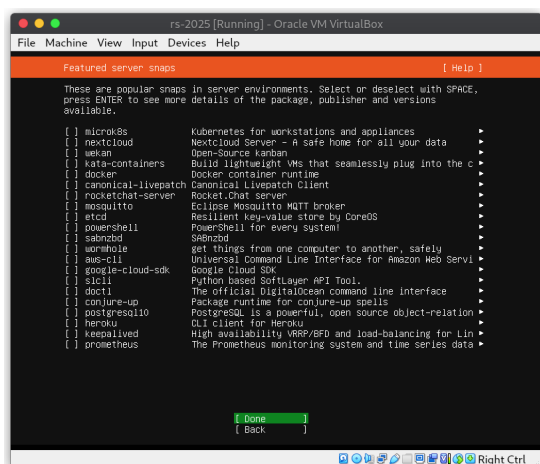
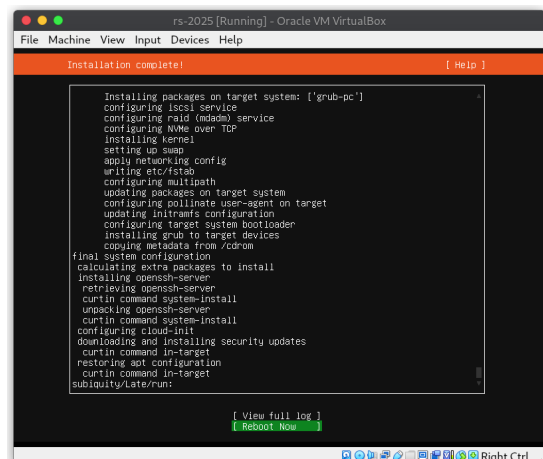


Figura 1.11.: Definición del perfil del usuario por defecto

8. Ignorar el siguiente diálogo y no instalar Ubuntu Pro
9. Seleccionar la instalación del servidor OpenSSH **sin** importar claves.
10. **No** instalar ningún *snap* adicional durante el proceso instalación (!)
11. Dejar que se realice la instalación y cuando aparezca en la pantalla la opción «Reboot now», seleccionarla para volver a arrancar la máquina.



(a) Último paso antes de la instalación. **No** seleccionar nada



(b) Arrancar la VM con el SO instalado

12. Dejar que la Máquina Virtual arranque y, una vez aparezca el mensaje de entrada en ella, entrar con el usuario `student`. Puede ser que aparezcan mensajes en el terminal. Estos mensajes se pueden ignorar (pulsando la tecla Enter).

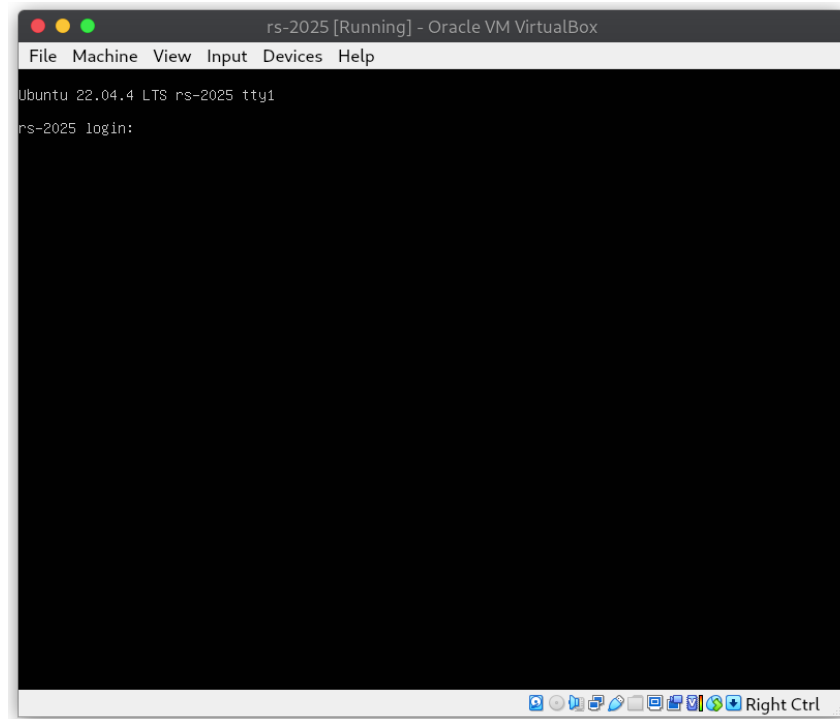


Figura 1.13.: Arranque en modo texto de la VM

1.2. Instalación del entorno gráfico

En este punto tiene instalado el SO básico. Vuelva a arrancar la VM. Si no se olvidó de quitar la imagen ISO del instalador, debería aparecer una terminal gráfica virtual, pidiendo el usuario y contraseña. Utilizando el usuario que definió anteriormente en el proceso de instalación, entre en la cuenta del usuario.

Para mayor estabilidad, refresque el índice del repositorio de paquetes y los paquetes instalados:

```
sudo apt update  
sudo apt upgrade -y
```

e instale el paquete `tasksel` para automatizar el proceso de instalación del entorno gráfico:

```
sudo apt install -y tasksel git  
sudo tasksel install mate-desktop
```

Este proceso podrá tardar algo de tiempo, puesto que requiere descargar una cantidad considerable de software de Internet.

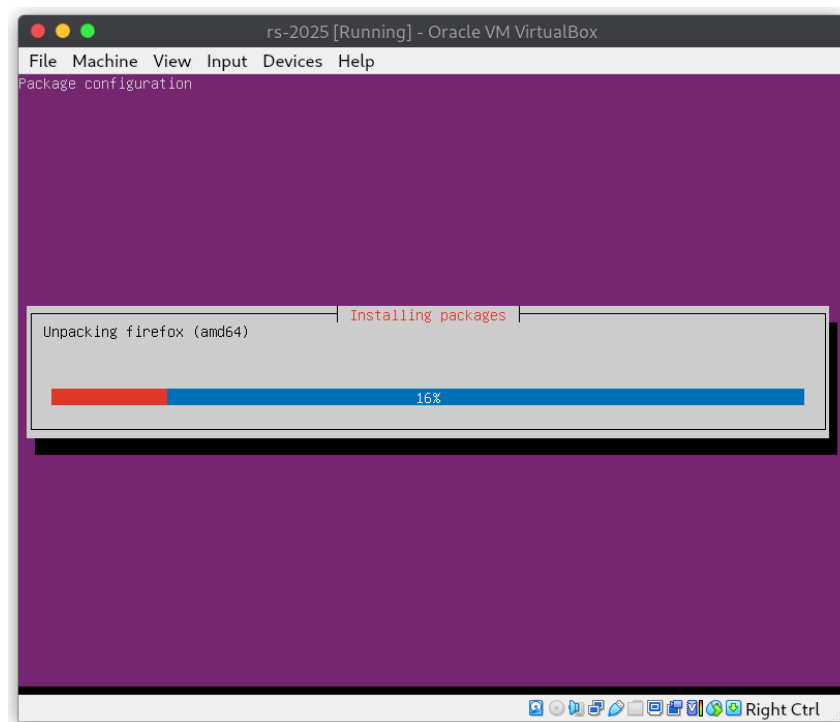


Figura 1.14.: Instalación del escritorio gráfico.

El paquete que más tiempo va a requerir en su instalación es el navegador Firefox. Una vez haya concluido, re arranque la VM desde la línea de comando:

```
sudo reboot -h now
```

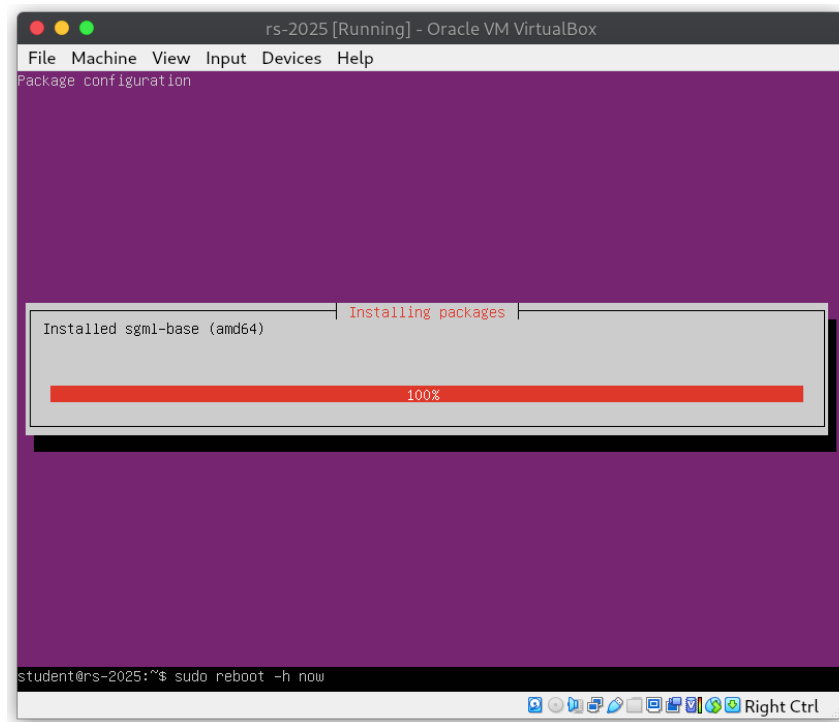


Figura 1.15.: Rearrancar la VM después de la instalación inicial

En el siguiente arranque, la VM arrancará en el entorno gráfico. En el primer arranque, es posible que aparezca un mensaje de error:

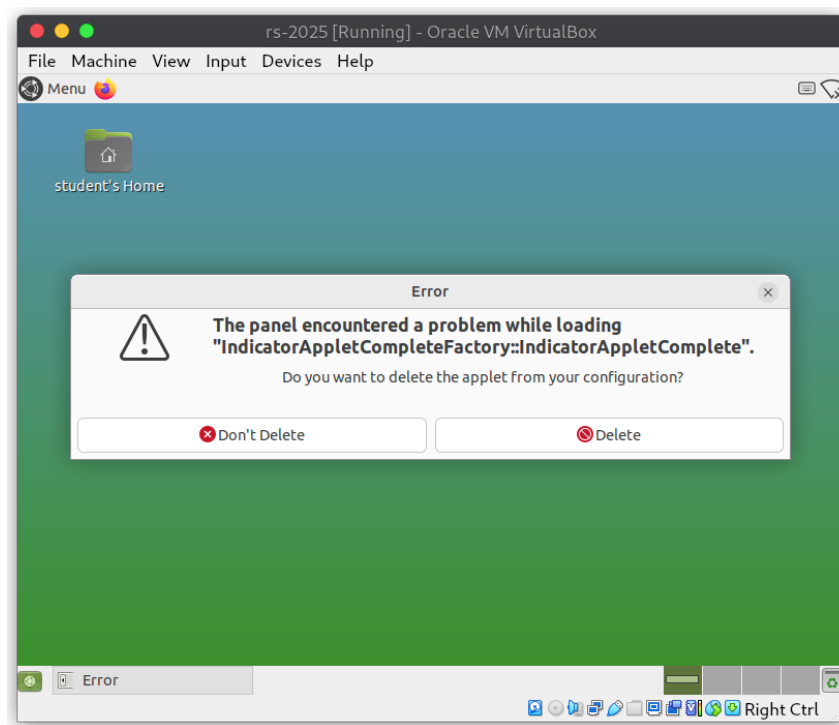


Figura 1.16.: Mensaje de error en el primer arranque

Seleccione **borrar** las *applets* que fallen para que no vuelva aparecer. Una vez esté en el entorno del escritorio gráfico, seleccione el menú principal de la esquina superior izquierda y vaya al centro de control (*Control Center*.) Seleccione la configuración de pantalla (*Displays*) y ajuste el tamaño de la pantalla virtual de acuerdo con la pantalla de su equipo:

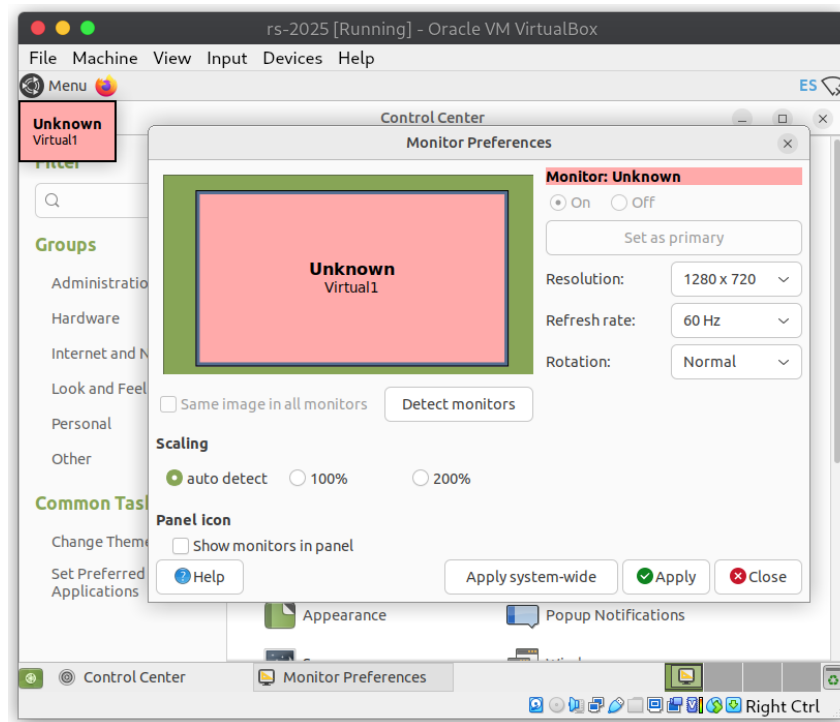


Figura 1.17.: Cambiando la resolución de la pantalla virtual.

Cuando encuentre un tamaño adecuado seleccione “*Keep this configuration*” y cierre la ventana del configurador de pantallas y del centro de control.

Vaya al menú y busque el terminal del MATE para continuar con la instalación:

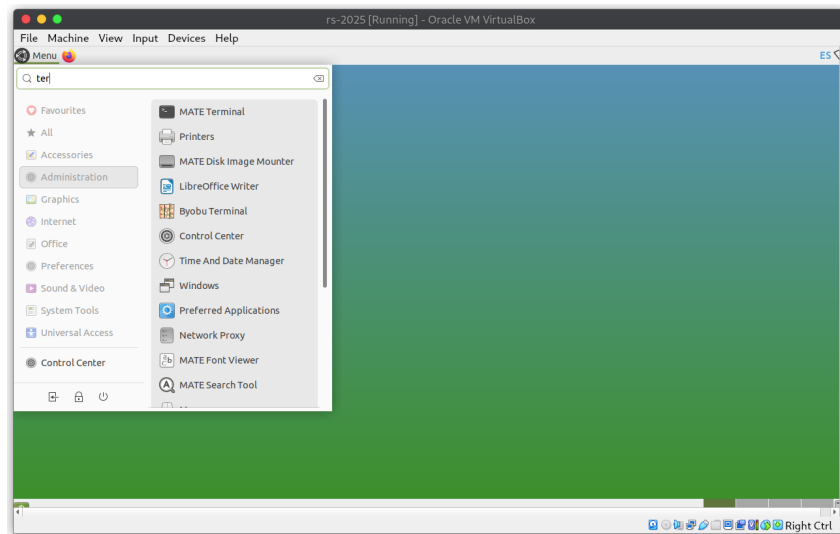


Figura 1.18.: Buscando el terminal MATE

Hito 1:

Se ha creado la VM con el sistema operativo y el entorno gráfico que se va a utilizar en el resto de la práctica, así como en las restantes prácticas del curso.

1.3. Instalación del entorno de emulación de redes OpenFlow

El entorno completo de emulación de redes OpenFlow consta de dos grandes paquetes: el emulador de redes *Mininet* y el controlador OpenFlow *Ryu*. Para Mininet utilizaremos la versión que nos proporciona Ubuntu.

```
sudo apt install -y mininet
```

Posteriormente, clone el repositorio de Mininet para disponer también del código fuente y de los ejemplos que le guiarán en posteriores desarrollos.

```
cd
git clone https://github.com/paaguti/mininet.git
```

Para el controlador Ryu, nos descargaremos una versión actualizada y la instalaremos utilizando las herramientas de gestión de software del lenguaje de programación Python3.

Empezaremos por instalar el gestor de software para Python3 y clonar el repositorio con el código fuente de Ryu:

```
sudo apt install -y python3-pip
cd
git clone https://github.com/paaguti/ryu.git
```

Y luego, instalaremos el programa propiamente dicho:

```
cd ~/ryu
pip install .
pip install -r tools/optional-requirements
```

Este proceso nos instala el software en un directorio oculto del usuario student. Para poder acceder a él, tendremos que incluirlo en la variable PATH:

```
echo 'export PATH=$HOME/.local/bin:$PATH' >> ~/.bash_aliases
```

Tenga cuidado y utilice la comilla sencilla ('). Las comillas dobles pueden no funcionar. Cierre el terminal y abra uno nuevo. Para comprobar la instalación, ejecute el comando:

```
which ryu-manager
```

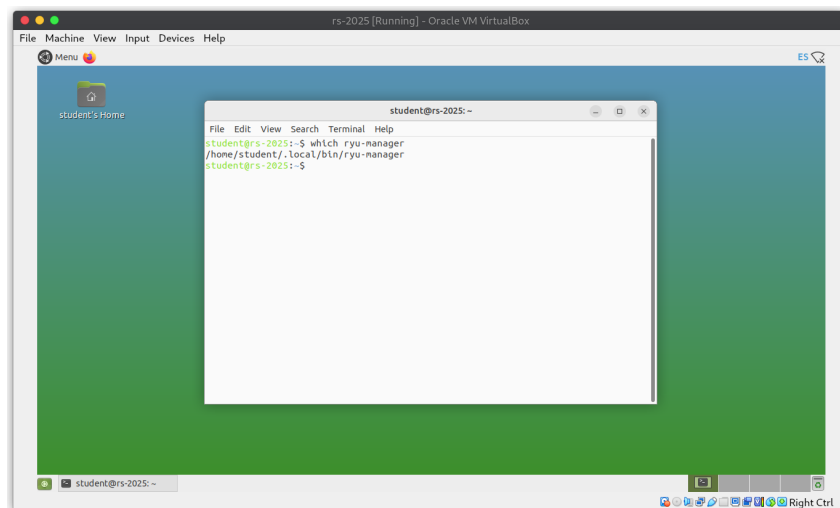


Figura 1.19.: Ryu correctamente instalado

Hito 2:

En este punto ha creado el entorno completo de emulación de redes OpenFlow que necesita para las restantes prácticas. Cree una imagen de la VM para poder importarla en un futuro si experimentara problemas y tuviera que empezar con una VM limpia.

Ajuste fino en entornos macOS y UTM

Si está instalando el entorno en una máquina Apple y está utilizando el hipervisor UTM, puede instalar el paquete `spice-vdagent` para poder utilizar el portapapeles de su equipo dentro de la VM[4]. Para ello, abra un terminal y ejecute el siguiente comando:

```
sudo apt install -y spice-vdagent
```

En la página de la referencia encontrará otras funcionalidades que pueden resultar interesantes a la hora de integrar la VM con su equipo.

2. Contenedores

Después de haber visto los procesos que permiten generar y gestionar Máquinas Virtuales, vamos a ver cómo trabajar con contenedores. Son una solución más ligera para ejecutar entornos aislados dentro de una máquina. Un ejemplo es el controlador de Open Source MANO (OSM), que se verá en otro de los laboratorios de la asignatura.

Empezaremos con la creación de contenedores utilizando el entorno de los Linux Containers (LXC), creando el mismo servidor Web que hemos creado dentro de la máquina virtual y, finalmente, lo replicaremos en el entorno Docker.

Sugerencia

Antes de empezar, es recomendable que esta parte la realice sobre una copia de la VM. Guarde la VM original para realizar las restantes prácticas de la asignatura.

2.1. Linux Containers

LXC permite la creación de contenedores, que ejecutan aplicaciones de manera aislada. Empezaremos por instalar el paquete base `lxc`, junto con el paquete de utilidades `lxc-utils` y el de plantillas `lxc-templates`, que permite acceder directamente a una extensa colección de plantillas para creación de contenedores:

```
sudo apt install -y lxc lxc-utils lxc-templates
```

En una instalación por defecto, la gestión de los contenedores necesita privilegios de root, que se obtienen con el programa `sudo`.

1. Empiece por instalar localmente la imagen de Alpine Linux que se va a ejecutar en el contenedor:

```
sudo lxc-create -t download -n alpine -- -a amd64 -d alpine -r 3.19 \
--flush-cache
```

En el proceso se verá como se bajan los componentes del contenedor (metadatos y el sistema de ficheros (rootfs)).

```
Downloading the image index
Downloading the rootfs
Downloading the metadata
The image cache is now ready
Unpacking the rootfs

---
You just created an Alpinelinux 3.19 x86_64 (20240117_13:00) container.
```

2. A continuación vamos a arrancar el contenedor en modo “detached”, es decir, en segundo plano y sin una consola asignada. Esto no impide que accedamos posteriormente al controlador mediante una consola.¹

```
sudo lxc-start -n alpine -d
```

3. Veamos el estado del contenedor que acabamos de crear con el comando:

```
sudo lxc-info alpine
```

Debería responder con algo parecido a lo siguiente:

```
student@uc3m:~$ sudo lxc-info alpine
Name:      alpine
State:     RUNNING
PID:       2183
IP:        10.0.3.82
Link:      vethxMJWNV
TX bytes:  1.62 KiB
RX bytes:  7.07 KiB
Total bytes: 8.69 KiB
```

4. Se puede obtener más información sobre los contenedores desplegados con el comando

```
sudo lxc-ls -f
```

Este comando tiene una salida similar a

NAME	STATE	AUTOSTART	GROUPS	IPV4	IPV6	UNPRIVILEGED
alpine	RUNNING	0	-	10.0.3.26	-	false

Observe que, además del estado se puede ver la dirección IP que se le ha asignado al contenedor

Empiece a obtener capturas de pantalla de los siguientes pasos para incluirlas en el documento de laboratorio

5. A continuación, accedemos al contenedor con el comando:

```
sudo lxc-attach -n alpine
```

Con ello accedemos con el usuario root del contenedor al contenedor y podremos instalar un servidor Web con los siguientes comandos:

¹La sugerencia del comando `lxc-create` lleva a acceder directamente y en primer plano al contenedor.

```
apk update
apk upgrade
apk add lighttpd
```

A continuación, se arranca el servidor Web con el comando:

```
rc-service lighttpd start
```

Intente acceder al servidor desde un navegador Web dentro del guest de Virtualbox. (Utilice la dirección IP que obtuvo del comando `lxc-ls -f`). Es normal que diga que no se puede acceder, porque no se ha instalado todavía ningún contenido.

6. Cree una página Web minimalista con el comando

```
echo "Mi nuevo servidor" | tee /var/www/localhost/htdocs/index.html
```

Y refresque el navegador. Tendría que ver el mensaje “Mi nuevo servidor Web” en el navegador.

7. A continuación procedemos a parar el contenedor (bien en una pestaña nueva del terminal, bien saliendo de la shell del contenedor con `exit` o pulsando las teclas `Ctrl-D`).

```
sudo lxc-stop -n alpine
```

y comprobamos que el contenedor ha sido detenido:

```
sudo lxc-ls -f
```

Debería imprimir algo similar a lo siguiente

NAME	STATE	AUTOSTART	GROUPS	IPV4	IPV6	UNPRIVILEGED
alpine	STOPPED	0	-	-	-	false

8. Finalmente destruiremos el contenedor para liberar recursos con el comando:

```
sudo lxc-destroy -n alpine
```

Hito 3:

Refleje las capturas de pantalla del terminal de los pasos 5 a 8 en su documento de laboratorio

2.2. Docker

Una manera eficaz de automatizar el proceso que se ha realizado en la sección anterior es utilizar Docker [5]. Este entorno tiene muchas posibilidades, incluyendo la composición de imágenes para crear aplicaciones complejas. En nuestro caso nos vamos a limitar a definir una imagen sencilla en la que se ejecute de manera automatizada el servidor Web que se creó manualmente en la sección anterior.

2.2.1. Preparación del entorno

Antes de nada, instalaremos el entorno Docker. Para este laboratorio utilizaremos el gestor de paquetes snap.

```
sudo snap install docker
```

Vamos a utilizar un directorio de desarrollo, en el que pondremos los ficheros de definición y otros ficheros que necesitemos. Se recomienda utilizar el directorio ~/Devel, de manera que si no existe, proceda a crearlo y si ya existe y tiene ficheros, guárdelos si los necesita y bórralos del directorio.

Empezaremos por crear un sencillo fichero que se llame index.html con el contenido de la página Web de prueba:

```
<h1>Hola gente!</h1>
<p>Copiado con el Dockerfile</p>
```



A continuación crearemos el fichero Dockerfile con el siguiente contenido:

```
FROM alpine:3.14.3
MAINTAINER Yo <student@localhost>
RUN apk update && apk upgrade && apk add lighttpd

COPY ./index.html /var/www/localhost/htdocs/

CMD ["/usr/sbin/lighttpd", "-f", "/etc/lighttpd/lighttpd.conf", "-D"]
EXPOSE 80/tcp
```



Empiece a tomar capturas de pantalla de los siguientes pasos para incluirlas en el documento de laboratorio.

1. Cree la imagen del contenedor, que referenciaremos como webserver:v1 utilizando el fichero Dockerfile que acaba de crear:

```
cd Devel
sudo docker build -t webserver:v1 .
```

El proceso se debería de reflejar de la siguiente manera en el terminal:

```
Sending build context to Docker daemon 73.73kB
Step 1/6 : FROM alpine:3.10.2
----> 961769676411
Step 2/6 : MAINTAINER Yo <student@localhost>
----> Using cache
----> f03cd97cbbde
Step 3/6 : RUN apk update && apk upgrade && apk add lighttpd
----> Using cache
----> 2fc13611f882
Step 4/6 : COPY ./index.html /var/www/localhost/htdocs/
----> Using cache
----> ec6467a9ce87
Step 5/6 : CMD ["/usr/sbin/lighttpd", "-f", "/etc/lighttpd/lighttpd.conf", "-D"]
----> Using cache
```

```
---> 9be92da5aab0
Step 6/6 : EXPOSE 80/tcp
---> Using cache
---> 460982efaebb
Successfully built 460982efaebb
Successfully tagged webserver:v1
```

2. A continuación instanciaremos la plantilla `webserver:v1` que acabamos de crear en un contenedor llamado `webserver`.

```
sudo docker run -d -it --name webserver webserver:v1
```

3. Después, comprobaremos que la instancia está ejecutando.

```
sudo docker ps -a
```

El resultado tendría que ser algo similar a

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
f18aa36f5d8e	webserver	"/usr/sbin/lighttpd ..."	26 seconds ago	Up 24 seconds
		optimistic_hodgkin		

Si algo va mal, habrá que ver si se ha creado el contenedor, pero el programa especificado en la línea CMD en el Dockerfile ha terminado de manera inesperada:

CONTAINER ID	IMAGE	PORTS	COMMAND	NAMES	CREATED
0fab1f558aa3	webserver		"/usr/sbin/lighttpd ..."	laughing_lederberg	16 seconds ago
Exited (0) 15 seconds ago					

4. Para poder acceder al servidor, hay que buscar la dirección IP que le ha sido asignada. Para ello, utilizar el comando:²

```
docker inspect webserver
```

Buscar la sección "NetworkSettings" y dentro de ella, la variable "IPAddress"

```
...
"NetworkSettings": {
  "Bridge": "",
  "SandboxID": "ad815ddbc1580b4148f1b826e0b658887db57c63ab038042bb06d3e0ca1474fe",
  "HairpinMode": false,
  "LinkLocalIPv6Address": "",
  "LinkLocalIPv6PrefixLen": 0,
  "Ports": {
    "80/tcp": null
  },
  "SandboxKey": "/var/snap/docker/423/run/docker/netns/ad815ddbc158",
  "SecondaryIPAddresses": null,
  "SecondaryIPv6Addresses": null,
  "EndpointID": "0d75ea6cf8aaa873fd6903c50b1fa31dc70185586636054b7e36f4650265b687",
  "Gateway": "172.17.0.1",
  "GlobalIPv6Address": "",
  "GlobalIPv6PrefixLen": 0,
  "IPAddress": "172.17.0.4",
  ...
}
```

²Cambiar `webserver` por el nombre del contenedor, si se ha utilizado otro nombre.

Y acceder a esa dirección desde el navegador. Debería aparecer la página `index.html` que hemos creado en el directorio `~/Devel`.

5. Finalmente, parar el contenedor y borrarlo. Para ello

Dar el comando:

```
sudo docker stop webserver
```

Compruebe que el contenedor está parado con

```
sudo docker ps -a
```

y luego bórralo con el comando:

```
sudo docker ps -a -f status=exited -q | xargs sudo docker rm
```

para borrar los contenedores que se hayan creado. Finalmente, se borran las imágenes instaladas con el comando:

```
sudo docker images -a -q | xargs sudo docker rmi
```

Alternativamente, una vez parado el contenedor del servidor Web, puede utilizar el comando

```
sudo docker system prune
```

para hacer una limpieza a fondo del entorno Docker. Esta opción hará aparecer el mensaje

```
WARNING! This will remove:
- all stopped containers
- all networks not used by at least one container
- all dangling images
- all dangling build cache
Are you sure you want to continue? [y/N]
```

Contestar y seguido de la tecla Enter

Hito 4:

Ha creado un contenedor Docker con un fichero de especificación y ha instalado y arrancado correctamente un servidor Web.

Muestre en el documento todos los pasos seguidos mediante capturas de pantalla. Compruebe el espacio de direccionamiento utilizado y compárelo con el espacio de direccionamiento de LXC.

2.2.2. Eliminar el snap de Docker

Para limpiar la VM, se tiene que desinstalar el snap de Docker y eliminar el estado asociado a él con los siguientes comandos:

```
student@uc3m:~/Devel$ sudo snap remove docker
docker removed
student@uc3m:~/Devel$ sudo snap saved
Set Snap    Age    Version  Rev   Size   Notes
1   docker  11.0s  20.10.24 2904  9.31kB auto
student@uc3m:~/Devel$ sudo snap forget 1
Snapshot #1 forgotten.
```

El comando `snap saved` se utiliza para obtener el índice del estado asociado al snap que hemos desinstalado y el `snap forget` lo elimina.

A. Ajustes en la imagen de VirtualBox

Estas instrucciones sirven para todas las prácticas de la asignatura **siempre que se realicen en nuestro ordenador **no** en los servidores del departamento**. Se recomienda utilizar una VM distinta para este laboratorio y los restantes laboratorios de SDN.

A.1. Instalación de las *Guest Additions*

Es altamente recomendable instalar las *Guest Additions*. Para ello, insertar el CDROM desde el menú de la VM cuando se arranque por primera vez. Seguir los menús y cuando pida el clave del usuario, poner 5tud3nt.

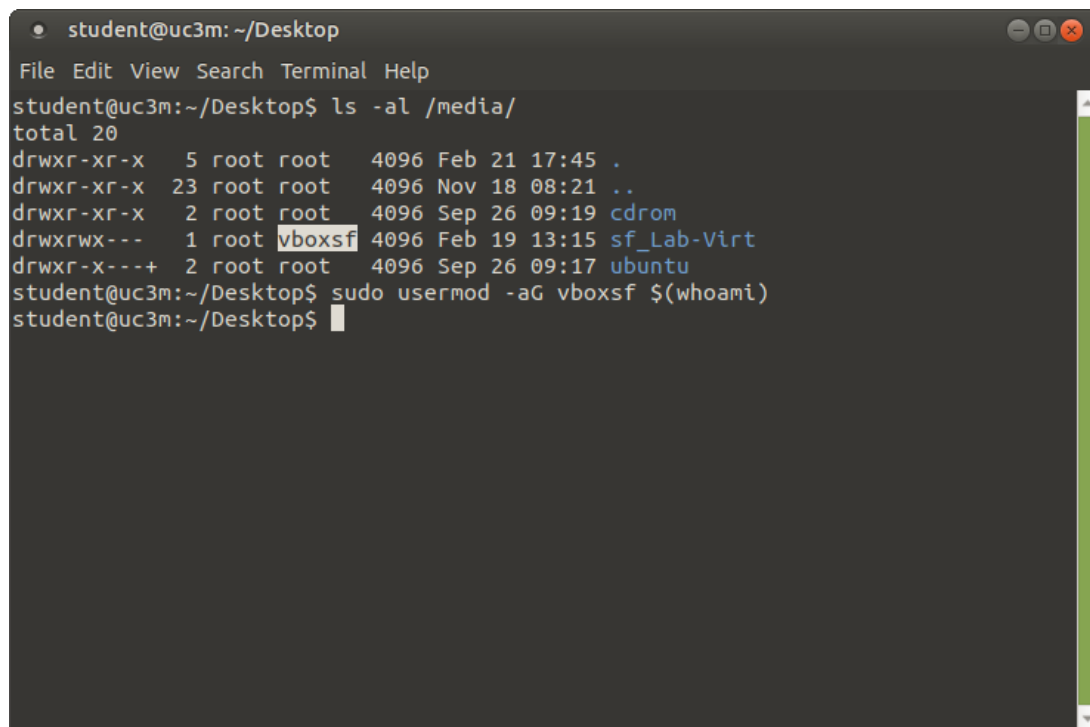
Una vez termine, expulsar el CDROM virtual y apagar la VM.

A.2. Mejoras antes del segundo arranque

Con la VM parada, en el diálogo de VirtualBox, cambiar la tarjeta gráfica a VBoxSVGA. Después aparecerá un mensaje de advertencia sobre la tarjeta gráfica que se puede ignorar.

En los dispositivos, añadir una carpeta compartida para poder transferir ficheros entre la VM y el ordenador físico.

Arrancar la VM y comprobar que la pantalla virtual se ajusta al tamaño de la ventana. Después abrir una ventana y añadir nuestro usuario al grupo vboxsf:



```
student@uc3m: ~/Desktop
File Edit View Search Terminal Help
student@uc3m:~/Desktop$ ls -al /media/
total 20
drwxr-xr-x  5 root root  4096 Feb 21 17:45 .
drwxr-xr-x 23 root root  4096 Nov 18 08:21 ..
drwxr-xr-x  2 root root  4096 Sep 26 09:19 cdrom
drwxrwx---  1 root vboxsf 4096 Feb 19 13:15 sf_Lab-Virt
drwxr-x---+  2 root root  4096 Sep 26 09:17 ubuntu
student@uc3m:~/Desktop$ sudo usermod -aG vboxsf $(whoami)
student@uc3m:~/Desktop$
```

Figura A.1.: Comprobar el grupo de la carpeta compartida y añadir al usuario en él

Nota En la captura, se seleccionó el nombre con el ratón, se copió al portapapeles con Ctrl-Shift-C y luego se pegó al comando con Ctrl-Shift-V.

Para evitar necesitar utilizar el comando sudo para la parte de Docker, se puede incluir al usuario en el grupo docker de manera análoga.

Finalmente, reiniciar la máquina.

A.3. Utilizar sudo sin necesidad de clave

Se puede permitir que el usuario student funcione sin necesidad de introducir la clave. Para ello hay que definir el comportamiento del usuario en el comando, creando un fichero en el directorio /etc/sudoers.d:

```
echo "student ALL=(ALL) NOPASSWD: ALL" | sudo tee /etc/sudoers.d/student
```

A partir de aquí habrá que poner especial cuidado cuando se utilice el comando sudo.

Acrónimos

CPU	procesador
LXC	LinuX Containers
SO	Sistema Operativo
OSM	Open Source MANO
SDN	Software Defined Networking
VM	Máquina Virtual

Referencias

- [1] *VirtualBox*. url: <https://www.virtualbox.org>.
- [2] Microsoft Corp. *Configure and manage Hyper-V*. Ene. de 2024. url: https://learn.microsoft.com/en-us/training/modules/configure-manage-hyper-v/?WT.mc_id=academic-89565-abartolo.
- [3] UTM, ed. *Securely run operating systems on your Mac*. Ene. de 2023. url: <https://mac.getutm.app/>.
- [4] *UTM: guest support - Linux*. 9 de oct. de 2023. url: <https://docs.getutm.app/guest-support/linux/>.
- [5] Docker Inc. *Docker Containers*. 2019. url: <https://www.docker.com>.