

MAQUINA VIRTUAL - VIRTUALBOX OVA

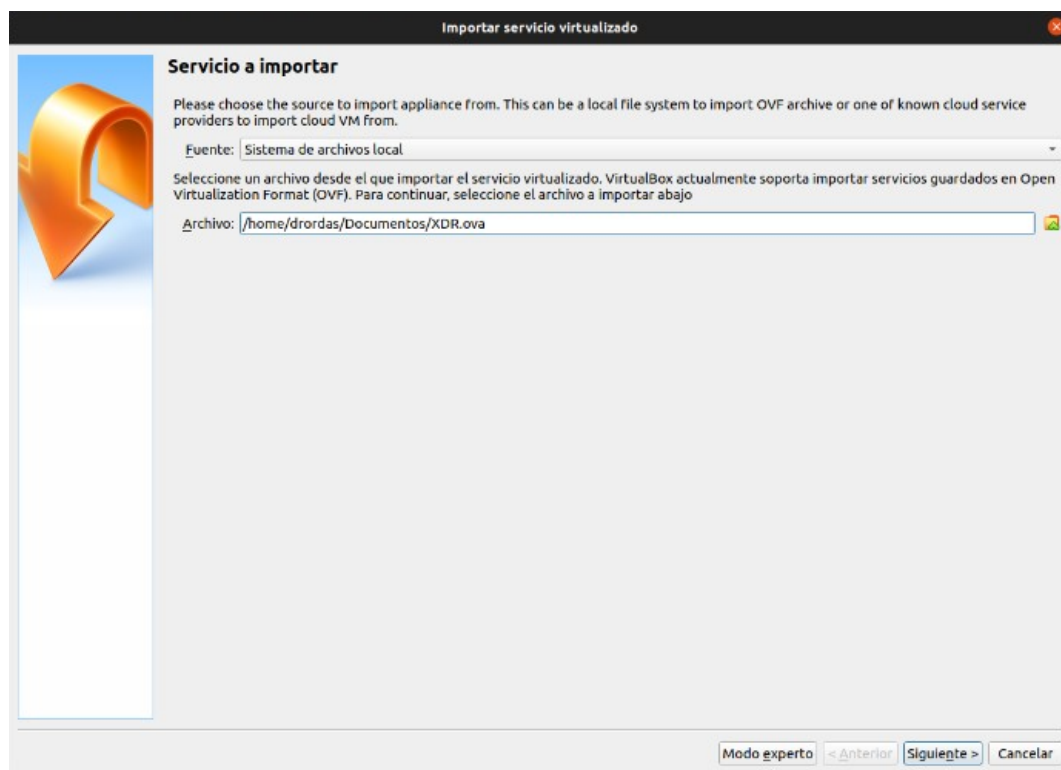
MANUAL DE FUNCIONAMIENTO

Descripción

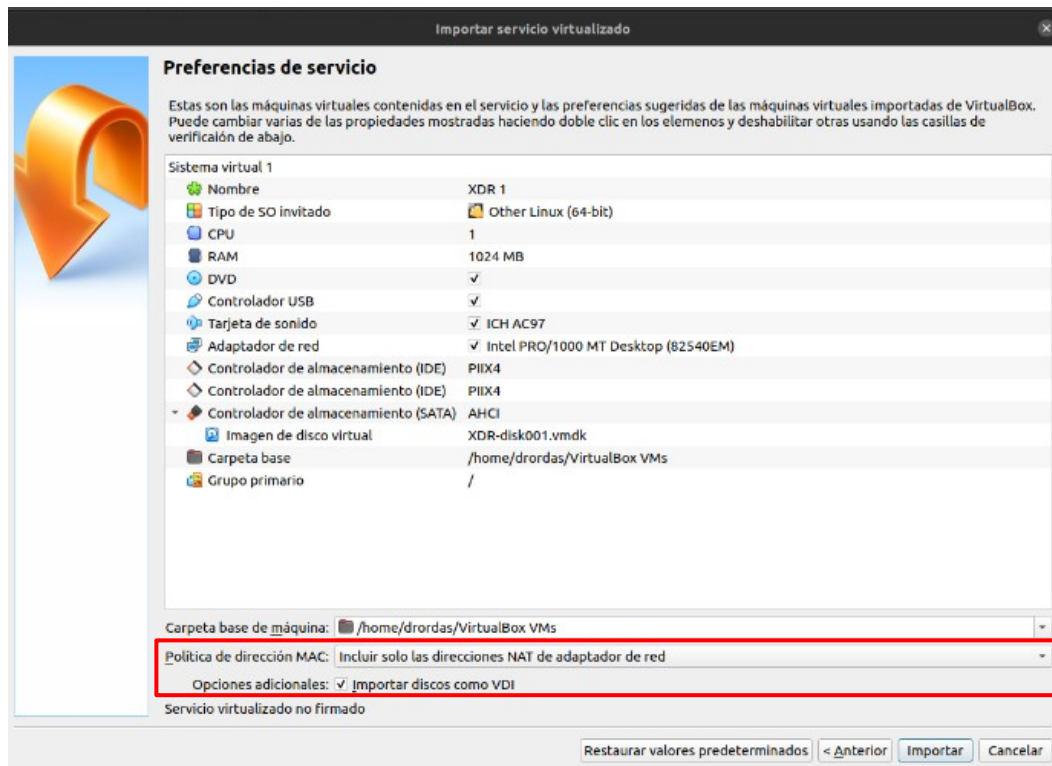
Para la realización de las prácticas del Tema I - Bases de Datos Relacionales, se proporciona un OVA de VirtualBox donde viene instalada una versión minimal de Ubuntu Server (sin interfaz gráfica) con las aplicaciones necesarias previamente instaladas (sshd y docker).

Paso 1 - Importación de la OVA

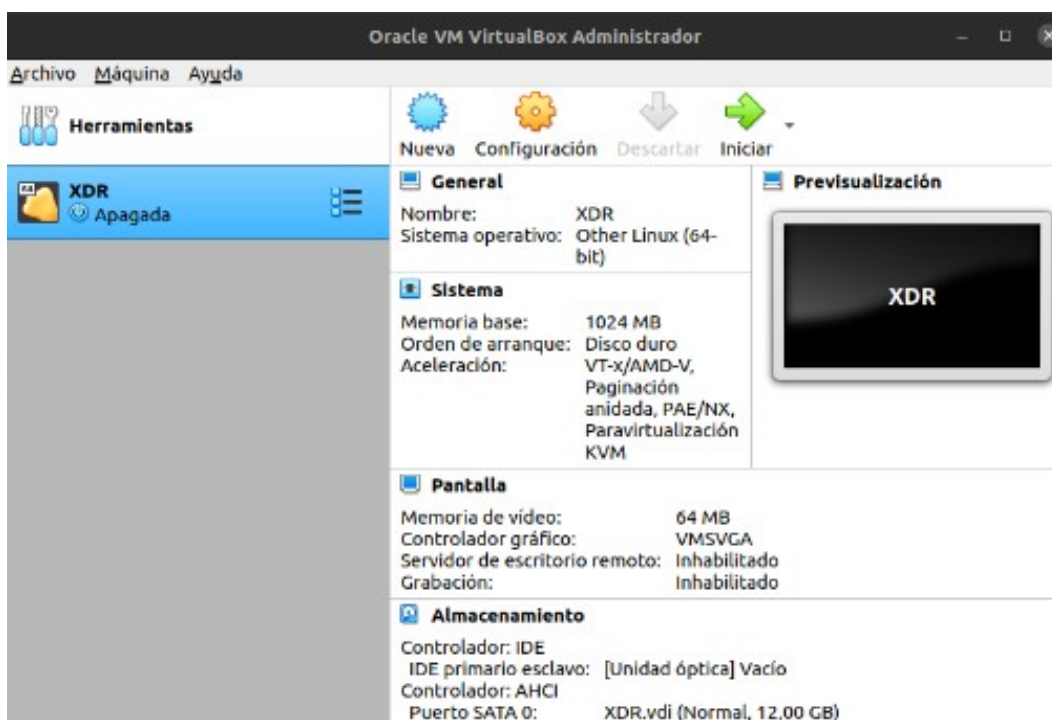
Para importar la OVA basta con ir a Archivo -> Importar Servicio Virtualizado (Control+I). En la pantalla “*Servicio a Importar*” hay que indicar en Archivo la OVA de la Máquina Virtual a importar (fichero XDR.OVA).



Una vez pulsado el botón de Siguiente, aparecerá la pantalla “*Preferencias del Servicio*” que mostrará los parámetros de configuración por defecto de la Máquina Virtual. En esta pantalla hay que tener en cuenta tres cuestiones: (i) no es recomendable modificar los parámetros que vienen por defecto ya que pueden ocasionar que la MV no funcione correctamente (ii) la *Política de Dirección MAC* debe estar en: “Incluir sólo direcciones NAT” y (iii) la opción de *Importar Disco como VDI* debe estar activa.

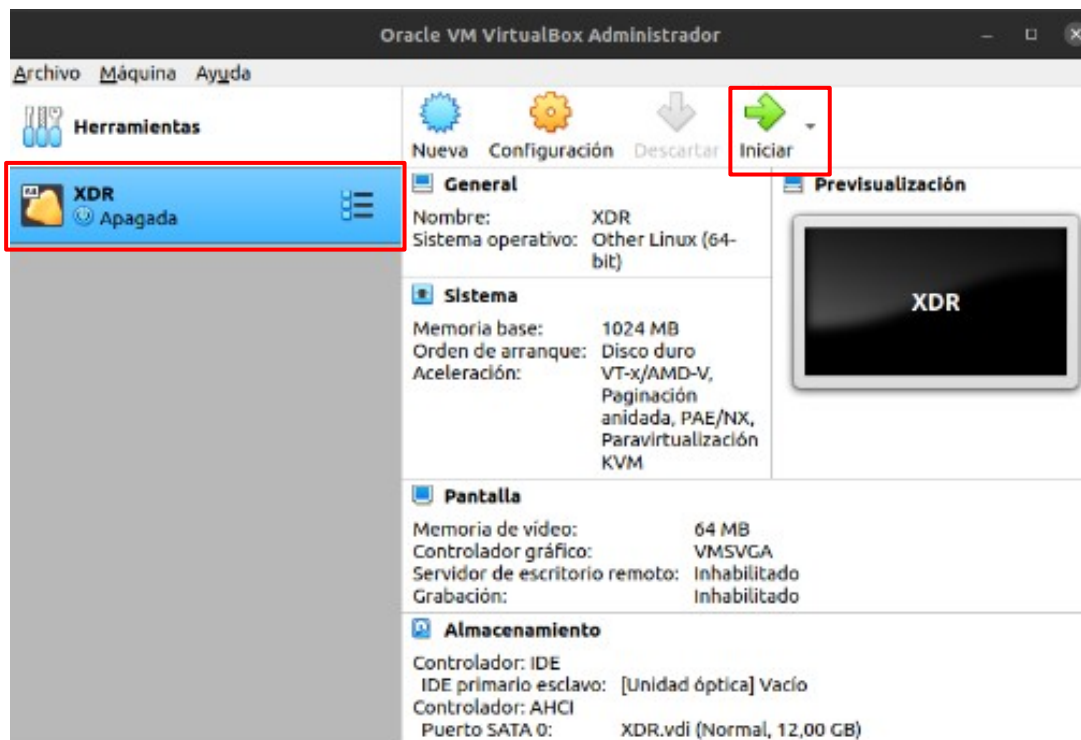


Finalmente, se pulsa sobre el botón Importar para terminar la importación de la Máquina Virtual. Una vez que el proceso de importación ha finalizado, la nueva Máquina Virtual debería aparecer en la pantalla de Administración de MV de Oracle.



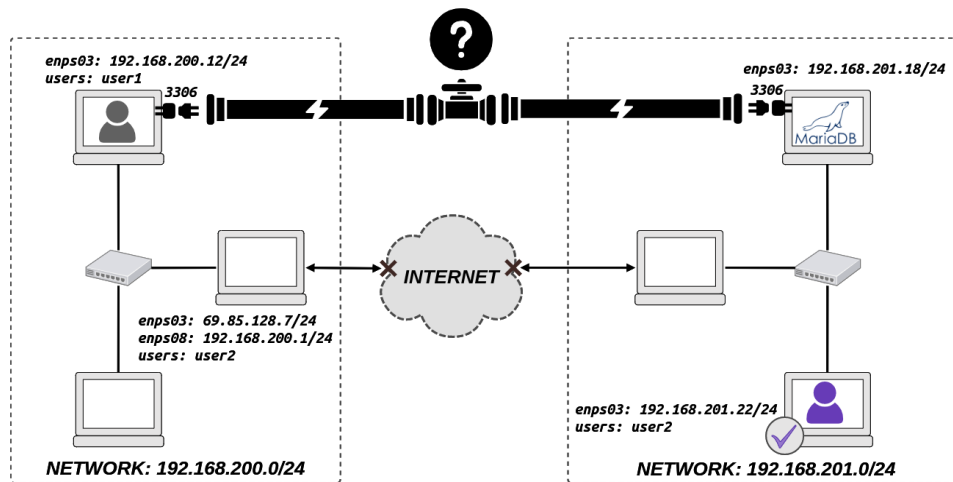
Paso 2 - Inicialización de la Máquina Virtual

Para iniciar la Máquina Virtual simplemente hay que seleccionarla y darle al botón de Iniciar.



Una vez iniciada la máquina, se abrirá una nueva ventana donde se cargará el SO Ubuntu. Al finalizar la carga (puede tardar un poco la primera vez), se mostrará la típica consola de Linux donde se pedirán las credenciales de acceso: user: **xdr**, pass: **xdr**.

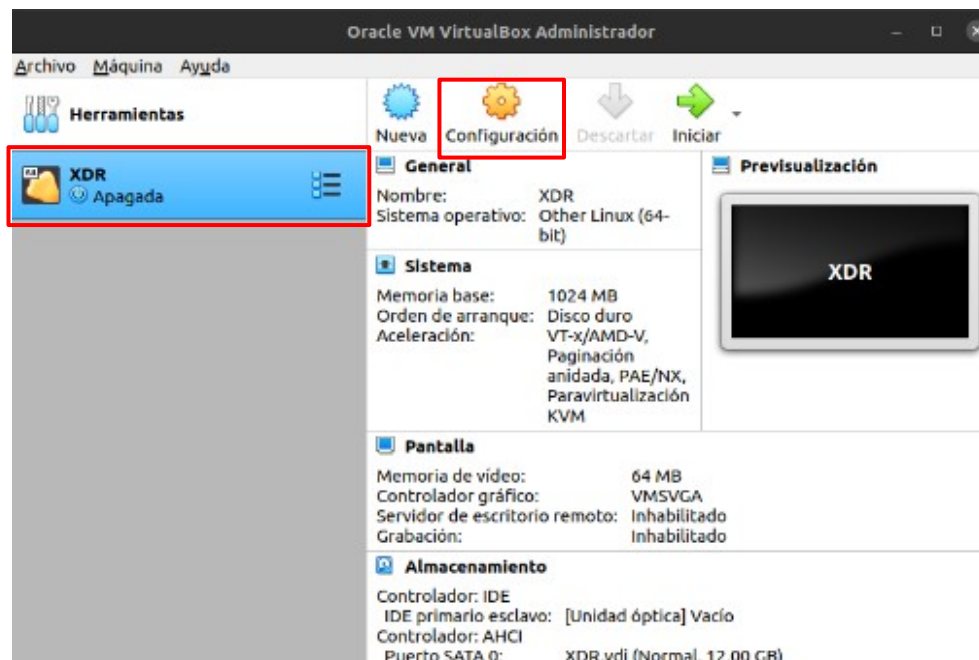
La consola que proporciona por defecto VB es muy incómoda ya que no permite ejecutar de manera sencilla comandos típicos como Cntrl+C, Cntrl+V, Cntrl+Z, Cntrl+Y,... . Por ello, se recomienda realizar la conexión en remoto mediante SSH usando una *terminal* del sistema. Sin embargo, usando la configuración por defecto de la MV esta tarea no se puede realizar ya que para poder tener acceso a Internet desde la propia Máquina Virtual, el adaptador de Red está configurado como NAT. Esto quiere decir que existe una traducción automática de las IPs de la red interna (invitado - 10.0.2.XX/24) con la red externa (anfitrión - 192.168.XXX.YYY). Este "mapeo" sólo se hace en una dirección (dentro-fuera) por lo que desde el equipo anfitrión NO se puede acceder remotamente al equipo invitado. Para solucionar este problema y poder conectarse remotamente mediante SSH usando la terminal al equipo invitado desde el equipo anfitrión es necesario realizar un SSH Remote Port-Forwarding. Esta técnica permite crear un túnel SSH mediante la conexión entre dos puertos (local y remoto). Básicamente consiste en conectar el puerto en remoto con uno en local (que se puede realizar porque el mapeo dentro-fuera está permitido). Así, cada vez que se realiza una conexión por el puerto local, se está comunicando directamente con el puerto en remoto.



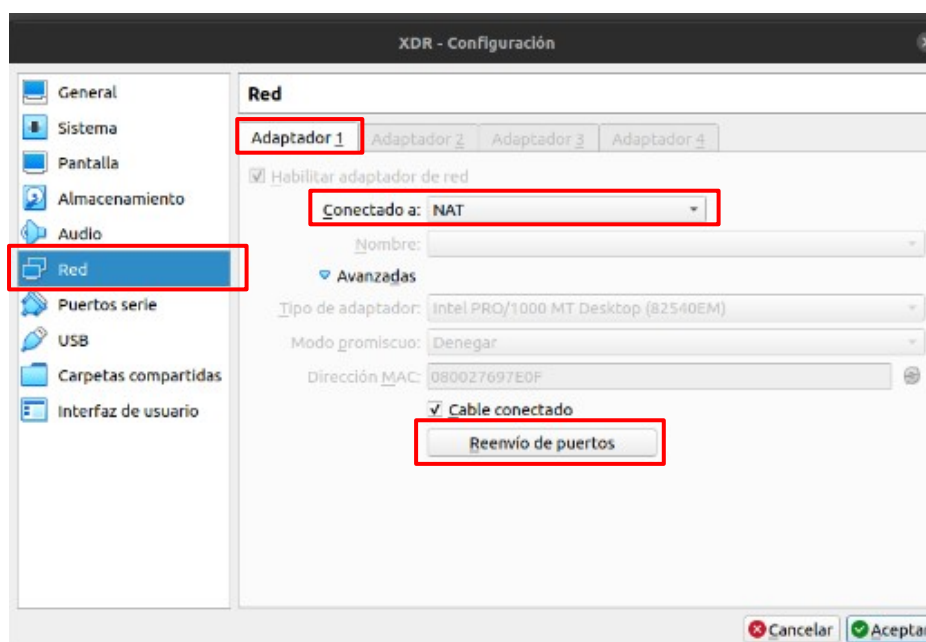
El Remote Port-Forwarding debería venir ya configurado en la OVA proporcionada, sin embargo, en ocasiones puede haber problemas de importación de la OVA que impida que se carguen correctamente todas las configuraciones.

Paso 3 - Configuración de Remote Port Forwarding

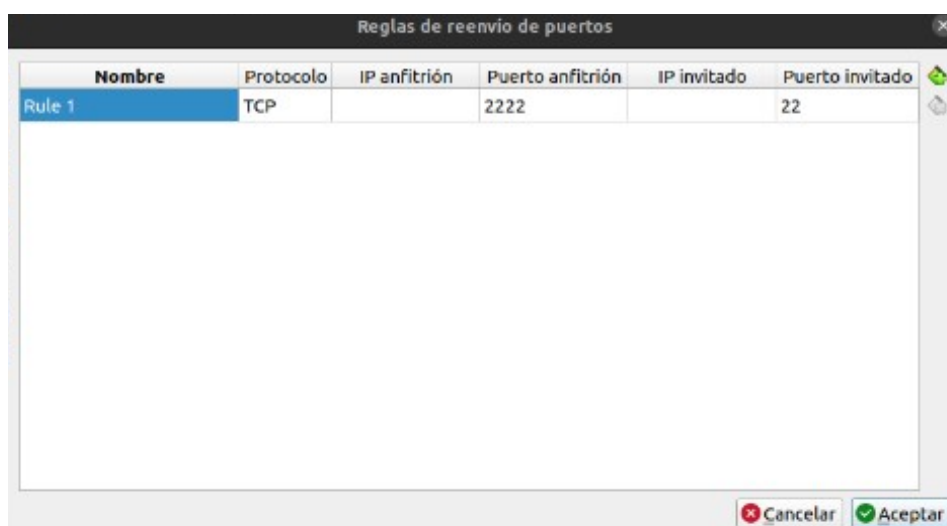
Para configurar el Remote Port-Forwarding hay que ir a la ventana de Administración de Oracle (ventana principal) y, teniendo la MV seleccionada, hay que pinchar sobre el botón de Configuración.



Con ello, se abrirá una nueva ventana que permite cambiar múltiples opciones de configuración de la MV. Para realizar el Remote Port Forwarding hay que ir al apartado de Red (menú vertical de la izquierda) y luego sobre la pestaña *Adaptador 1*, pinchar sobre el botón *Reenvío de Puertos*. Además, es importante comprobar que el modo de conexión sea NAT. Si no está como NAT, hay que cambiarlo.



A continuación, se abrirá una nueva ventana denominada “*Reglas de Reenvío de Puertos*” donde se definirá la regla que usaremos para poder acceder a la máquina invitada. El demonio SSH instalado en la máquina invitado está configurado para que escuche en el puerto 22 (puerto por defecto). Por tanto hay que definir una regla que conecte un puerto de la máquina anfitrión con el puerto 22 en la máquina invitada. Cuidado, se puede elegir un puerto aleatorio pero hay que tener en cuenta dos restricciones: (i) tiene que ser mayor que 1024 (los otros están reservados por el sistema) y solo se pueden usar si eres root (Linux) y (ii) no puede estar siendo ocupado por otro servicio.



Tal y como se puede observar en la imagen anterior, se ha establecido la regla 2222:22 pero se podría usar otro puerto siempre que cumpla con las restricciones indicadas anteriormente. Si se usa Linux (eso espero), no es recomendable hacer el “mapeo” con el puerto 22 del ordenador anfitrión ya que es muy común que esté ya ocupado ese puerto con el servidor SSH que suelen incluir por defecto las distribuciones Linux.

Una vez definida la regla, es importante darle al botón de *Aceptar* para que se realicen los cambios. Este proceso se puede realizar con la MV encendida y no es necesario reiniciarla. Sin embargo, si se usan SO Windows es recomendable hacerlo con ella apagada porque no suele ser capaz de refrescar correctamente la configuración cuando se realizan cambios sobre MV que ya se están ejecutando.

Paso 4 - Acceso remoto a la MV

Una vez que se ha configurado el Remote Port-Forwarding ya se puede acceder a la MV a través de un túnel SSH. Para ello, hay que abrir una terminal y ejecutar el siguiente comando:

```
ssh -p <puerto> user@direccion
```

Teniendo en cuenta que hay un port-forwarding en el puerto 2222 y que el usuario de la MV es xdr entonces el comando sería:

```
ssh -p 2222 xdr@localhost
```

La dirección es localhost porque lo que se está diciendo en el comando es, conéctate al puerto 2222 de mi propio equipo con el usuario xdr. Como en realidad el puerto 2222 está conectado al 22 (que es donde escucha el SSH) entonces lo que realmente se está haciendo es realizando la petición SSH al puerto 22 del host remoto (invitado).

```
drordas@work:~$ ssh -p 2222 xdr@localhost
The authenticity of host '[localhost]:2222 ([127.0.0.1]:2222)' can't be established.
ECDSA key fingerprint is SHA256:AwW0TLEI8S+AS7t1WutRc1R25icNSCYmn07I0BGDv0o.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Failed to add the host to the list of known hosts (/home/drordas/.ssh/known_hosts).
xdr@localhost's password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-104-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of lun 14 mar 2022 17:49:17 UTC

System load: 0.0          Users logged in: 0
Usage of /: 53.1% of 11.75GB IPv4 address for br-55a0c9d99bb6: 172.18.0.1
Memory usage: 25%         IPv4 address for docker0: 172.17.0.1
Swap usage: 0%            IPv4 address for enp0s3: 10.0.2.15
Processes: 108

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

https://ubuntu.com/blog/microk8s-memory-optimisation

19 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Last login: Mon Mar 14 15:41:49 2022 from 10.0.2.2
xdr@xdr:~$
```

Paso 5 - Manipulación de contenedores

Para la realización de la práctica se proporcionan dos aplicaciones, un Sistema Gestor de Bases de Datos (MySQL 5.7.28) y un Administrador de Bases de Datos (PHPMyAdmin). Con el fin de facilitar el manejo y control de las aplicaciones se ha decidido instalar cada una de ellas en un contenedor Docker. Por otro lado, como las aplicaciones deben estar unidas (PHPMyAdmin debe tener acceso a MYSQL) se proporciona un fichero docker-compose.yml en el directorio home del usuario xdr (/home/xdr). La estructura del fichero se muestra a continuación:

```

01 version: '3'
02
03 services:
04   mysql:
05     image: mysql:5.7.28
06     ports:
07       - 3306:3306
08     environment:
09       - MYSQL_ROOT_PASSWORD=root
10       - MYSQL_DATABASE=database
11       - MYSQL_USER=user
12       - MYSQL_PASSWORD=password
13     volumes:
14       - mysql_data:/var/lib/mysql
15
16   phpmyadmin:
17     image: phpmyadmin/phpmyadmin
18     ports:
19       - 8080:80
20     environment:
21       - PMA_HOST=mysql
22     depends_on:
23       - mysql
24
25 volumes:
26   mysql_data:

```

Tal y como se puede observar, la línea 07 define el puerto de escucha del MySQL en este caso se está diciendo que el puerto 3306 de fuera del contenedor se mapee con el puerto 3306 de dentro del contenedor (que es en el que escucha MySQL por defecto). En cuanto a las líneas 09-12, definen la configuración de MySQL, concretamente indican las credenciales de acceso (usuario: root, password: root) y el nombre de la base de datos que se usará por defecto (en ese caso database). Por otro lado, la línea 14 define el path donde se va a ubicar la base de datos (en este caso /var/lib/mysql). Es muy importante que la información se mapee fuera del contenedor, sino al parar/eliminar el contenedor se perderá toda la información.

En lo que respecta a PHPMyAdmin, cabe destacar las líneas 19 y 21. La primera indica que el puerto 8080 del contenedor se mapee con el puerto 80 dentro del contenedor ya que es donde escucha PHPMyAdmin por defecto. Por otro lado, la línea 21 se encarga de enlazar PHPMyAdmin con MySQL. Para concluir, se considera relevante destacar la línea 22 que se encarga de crear una dependencia que impida que se cargue PHPMyAdmin si no se ha cargado totalmente MySQL.

En resumen, docker-compose permite administrar de una manera sencilla (i) los parámetros de configuración de cada contenedor, (ii) los puntos de enlaces entre ellos (si los hay) y (iii) las dependencias entre los mismo (si las hay).

Trabajar con docker-compose es muy sencillo y aunque tiene muchas opciones distintas, nos centraremos en las cuatro que pueden ser necesarias para la administración de los contenedores definidos en el fichero docker-compose.

docker-compose	start	Inicializar servicios
	stop	Parar los servicios
	up -d	Crear e inicializar los contenedores en segundo plano (-d)
	down	Parar y borrar los contenedores

Para ejecutar el comando docker-compose es importante tener en cuenta dos cuestiones (i) la necesidad de ser root (la cuenta xdr ya está en el fichero /etc/sudoers por lo que tiene privilegios de root usando la password xdr), y (ii) debe ejecutarse en el path donde está el fichero o bien especificarlo mediante el argumento -f <path_to_dockerfile>.

Es muy importante tener en cuenta que la MV por defecto no dispone de las imágenes docker necesarias para realizar la práctica (MySQL y PhpMyAdmin) por lo que debe ejecutarse obligatoriamente el comando *docker-compose up -d*. Este comando, bajará las imágenes y luego creará los contenedores con la configuración especificada en el fichero *docker-compose.yml*. La imagen que se incluye a continuación muestra la salida producida al ejecutar el comando por primera vez. Tal y como se puede observar, primero se descarga la imagen de MySQL y se verifica que sea correcta (comprobación mediante *digest*) y luego se descarga la imagen de PhpMyAdmin y se comprueba su correctitud. Una vez que ambas imágenes se han descargado correctamente, se procede a la creación de los contenedores (ver últimas dos líneas).


```
xdr@xdr:~$ sudo docker-compose up -d
Creating network "xdr_default" with the default driver
Pulling mysql (mysql:5.7.28)...
5.7.28: Pulling from library/mysql
804555ee0376: Pull complete
c53bab458734: Pull complete
ca9d72777f90: Pull complete
2d7aad6cb96e: Pull complete
8d6ca35c7908: Pull complete
6ddae009e760: Pull complete
327ae67bbe7b: Pull complete
31f1f8385b27: Pull complete
a5a3ad97e819: Pull complete
48bede7828ac: Pull complete
380afa2e6973: Pull complete
Digest: sha256:b38555e593300df225daea22aeb104eed79fc80d2f064fde1e16e1804d00d0fc
Status: Downloaded newer image for mysql:5.7.28
Pulling phpmyadmin (phpmyadmin/phpmyadmin)...
latest: Pulling from phpmyadmin/phpmyadmin
5eb5b503b376: Pull complete
8b1ad84cf101: Pull complete
38c937dadb7: Pull complete
6a2f1dc96e59: Pull complete
f8c3f82c39d4: Pull complete
90fc6462bd8e: Pull complete
c670d99116c9: Pull complete
268554d6fe96: Pull complete
6c29fa0d4492: Pull complete
73e23c50a259: Pull complete
81ac13c96fc2: Pull complete
b60a3e623949: Pull complete
dac5dd67fd59: Pull complete
fd46866d9c36: Pull complete
443a80ef4c80: Pull complete
5e6049224f95: Pull complete
213e66cdf56: Pull complete
9b9b44731108: Pull complete
Digest: sha256:1d3f6ec798a9faa201ada6678797f399202928e677dfccbb698ef0b511c798b0
Status: Downloaded newer image for phpmyadmin/phpmyadmin:latest
Creating xdr_mysql_1 ... done
Creating xdr_phpmyadmin_1 ... done
xdr@xdr:~$
```

Una vez que se han bajado las imágenes y creado los contenedores, el resto de las veces valdría con ejecutar *docker-compose start* (también sería válido ejecutar *docker-compose up -d*).

```
xdr@xdr:~$ sudo docker-compose up -d
Starting xdr_mysql_1 ... done
Starting xdr_phpmyadmin_1 ... done
xdr@xdr:~$
```

Hay que tener en cuenta que ambos contenedores deben estar en ejecución para que se pueda acceder y manipular la Base de Datos. Para gestionar los contenedores se pueden usar los siguientes comandos:

docker	ps [-a]	Lista [todos] los contenedores
	stop <id>	Detiene el contenedor con el ID especificado
	start [-d] <id>	Inicializa [en segundo plano] el contenedor con el ID concreto
	images	Muestra todas las imágenes docker existentes
	rm <id>	Eliminar un contenedor (debe estar detenido)
	rmi <id>	Eliminar una imagen concreta.

A continuación se muestra un ejemplo del uso y así como la salida obtenida de algunos de los comandos descritos en la tabla anterior.

```
xdr@xdr:~$ sudo docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
9edf5b210586	phpmyadmin/phpmyadmin	"/docker-entrypoint..."	34 minutes ago	Up 34 minutes	0.0.0.0:8080->80/tcp, :::8080->80/tcp
20beb279bcag	mysql:5.7.28	"docker-entrypoint.s..."	34 minutes ago	Up 34 minutes	0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp

```
xdr@xdr:~$
```

La imagen superior muestra los contenedores que están creados en Máquina Virtual gracias al uso del comando *docker ps -a*. En este caso todos los contenedores se están ejecutando STATUS UP, pero podría haber contenedores que estén parados.

```
xdr@xdr:~$ sudo docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
phpmyadmin/phpmyadmin	latest	5682e7556577	4 weeks ago	524MB
mysql	5.7.28	db39680b63ac	2 years ago	437MB

```
xdr@xdr:~$
```

En esta imagen se muestra la salida del comando *docker images*. Tal y como se indicó en la tabla superior, este comando permite mostrar las imágenes que se han descargado en la MV junto con su Identificador (único) y el tamaño que ocupa en disco.

Paso 6 - Acceso al Administrador de Bases de Datos PhpMyAdmin

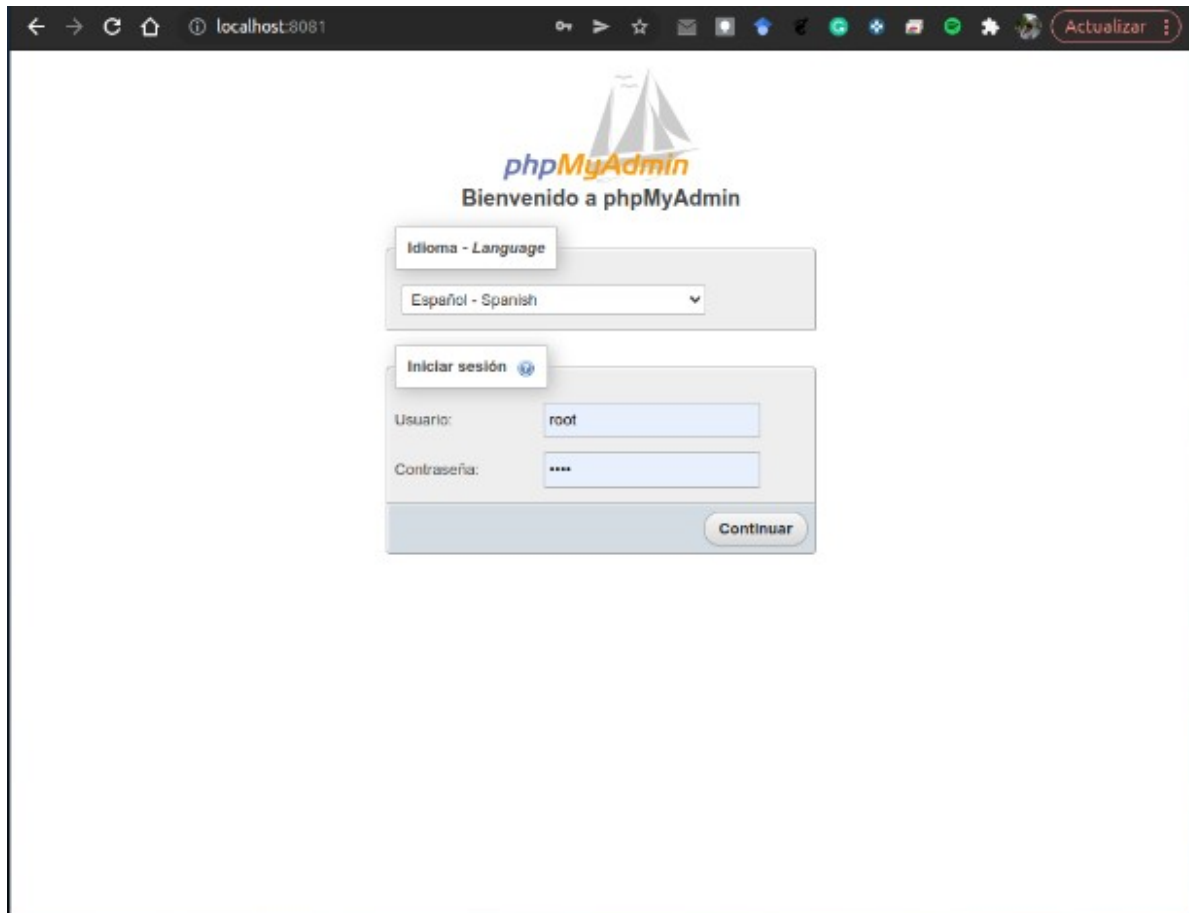
Tal y como se mostró en el *docker-file.yml* el contenedor de PhpMyAdmin escucha en el puerto 8080 (8080:8080). Sin embargo, tal y como se comentó en el Paso 4, no se puede acceder a la red del equipo invitado desde el anfitrión. Por tanto, hay que indicar en la configuración de red de la MV otra regla de reenvío de puertos que mapee un puerto de la máquina local (anfitrión) con el 8080 de la máquina invitada.



Tal y como se muestra en la imagen superior se ha mapeado el puerto 8081 de la máquina anfitrión con el puerto 8080 de la máquina invitada. Podría hacerse perfectamente con cualquier otro puerto libre (como el propio 8080 del anfitrión). La razón de usar el 8081 en este caso radica en que en el equipo anfitrión el puerto 8080 ya está ocupado con otro servicio.

Tal y como se comentó al principio de la guía, PhpMyAdmin es una aplicación Web que permite administrar de una manera intuitiva las Bases de Datos relacionales. Como es una aplicación web, la manera de conectarse es usando el protocolo HTTP (protocolo de capa

de aplicación que va encapsulado sobre un TCP). Por ello, para acceder a la aplicación basta con abrir un navegador e introducir la dirección localhost:8081 donde aparecerá la página de inicio.



Las credenciales para acceder al Sistema Gestor de Bases de Datos son usuario: root y password:root.