

Administración de sistemas UNIX - Proyecto práctico

Script maestro para la configuración de un cluster Linux

Curso 2021/2022

Introducción

El presente proyecto práctico tiene como objetivo ahondar en los conocimientos adquiridos por el alumno a lo largo de la asignatura a través de un caso aplicado. Partiendo de un análisis del problema propuesto, los alumnos deberán seleccionar razonadamente las herramientas y técnicas que un determinado sistema informático requiere para su administración. Posteriormente, deberán implementarlas y comprobar su correcto funcionamiento.

El caso propuesto en este proyecto es el de un *cluster* formado por múltiples máquinas ejecutando un sistema Linux. La tarea a realizar será, por tanto, la de elaborar un conjunto de *scripts* que faciliten al administrador el proceso de configuración de todo el sistema de manera centralizada. En concreto, los *scripts* permitirán llevar a cabo la configuración de dispositivos de almacenamiento, así como las partes servidor y cliente de servicios típicos de un *cluster*.

Normas básicas

- **Grupos:** el trabajo se realizará en grupos de tres alumnos.
- **Entrega:** 22 de diciembre de 2021. Se realizará a través de Moodle, en el apartado habilitado a tal efecto.

1. Preparaciones previas y entorno de trabajo

Para la realización del presente proyecto, este documento viene acompañado de una máquina virtual VirtualBox preconfigurada con el sistema operativo Ubuntu 20.04. Para simular el *cluster* a administrar, será necesario lanzar múltiples copias de dicha máquina virtual sobre una misma red virtual que las interconecte. Si bien podría emplearse cualquier otro sistema Linux, se recomienda encarecidamente hacer uso de la máquina virtual provista, puesto que la evaluación del proyecto se realizará exclusivamente sobre esta. Este apartado pretende dar algunas indicaciones en relación a dicha máquina virtual, su configuración y su puesta en marcha.

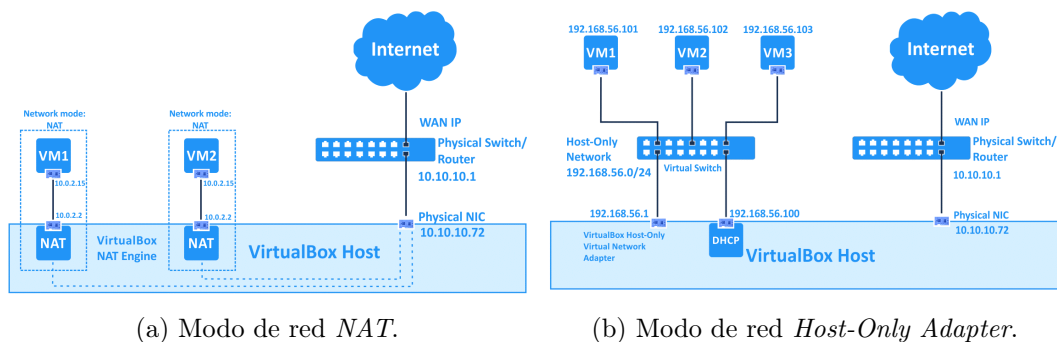


Figura 1: Esquemas de red utilizados en las instancias del proyecto².

1.1. Configuración inicial

El primer paso será la instalación del software VirtualBox¹, el cual se encuentra disponible de manera gratuita para múltiples plataformas. Dado que es posible encontrar abundante información en la red sobre cómo llevar a cabo su instalación, este documento asumirá que VirtualBox ya se encuentra listo en el sistema. Cabe señalar, no obstante, que en ocasiones resulta necesario cambiar la configuración de la máquina anfitriona para habilitar la tecnología de virtualización de la CPU, lo que permite mejorar el rendimiento de las máquinas virtuales. Del mismo modo, también se recomienda la instalación del denominado VirtualBox Extension Pack, que es necesario descargar aparte, y que proporciona funcionalidad adicional. La versión sobre la cual se ha probado la máquina virtual es la 6.1.26.

Una vez instalado y configurado, y antes de añadir la máquina virtual, resulta necesario configurar la red sobre la cual se ejecutarán las distintas instancias del *cluster*. En este sentido, VirtualBox ofrece diversos modos de configuración de los adaptadores de red de las máquinas virtuales. En particular, los dos modos que nos resultan de mayor interés para este proyecto son *NAT* y *Host-Only Adapter*:

- El modo *NAT*, representado en la Fig. 1a, es el modo por defecto, y permite a la máquina invitada acceder al exterior usando el servicio NAT a través de la interfaz de la máquina anfitriona. No obstante, este modo no permite comunicar máquinas virtuales entre sí, ni a la máquina anfitriona contactar con la máquina virtual, salvo mediante redirección de puertos.
- Por su parte, el modo *Host-Only Adapter*, mostrado en la Fig. 1b, posibilita la comunicación desde la máquina anfitriona hacia las máquinas virtuales a través de un adaptador creado a propósito para tal efecto, y además permite la comunicación de todas las máquinas virtuales que se adhieran a ella.

¹Página principal de VirtualBox: <https://www.virtualbox.org/>

²Imágenes obtenidas de <https://www.nakivo.com/blog/virtualbox-network-setting-guide/>

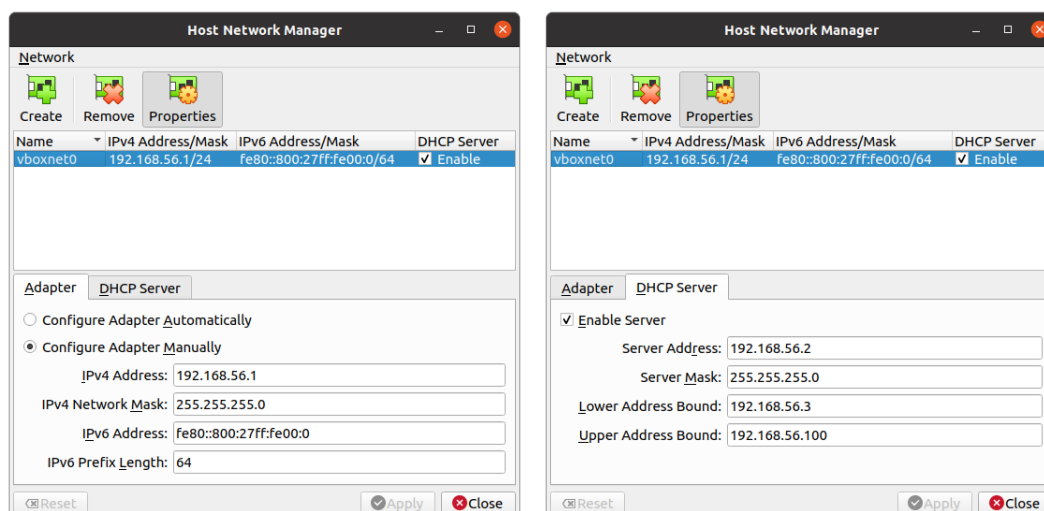


Figura 2: Parámetros de configuración del adaptador a usar en modo *Host-Only Adapter*.

Así, usando ambos modos se consigue comunicación bidireccional entre anfitrión y cada máquina virtual, y entre máquinas virtuales. Los pasos a seguir para crear el adaptador de red utilizado por el modo *Host-Only Adapter* son los siguientes:

1. En la ventana principal de VirtualBox, hacer clic sobre *File > Host Network Manager...*
2. En el cuadro que se mostrará, si no existe ningún adaptador creado por defecto, pulsar sobre el botón *Create*. En este sentido, es muy probable que el proceso de instalación de VirtualBox haya creado el adaptador `vboxnet0` automáticamente, el cual podemos utilizar sin necesidad de crear uno adicional.
3. Configurar adecuadamente los valores de red asociados al adaptador. En particular, si bien no resulta estrictamente necesario, resulta importante desde un punto de vista práctico asegurarnos de que la opción *Enable Server* de la pestaña *DHCP Server* se encuentra marcado para activar el servicio DHCP y que VirtualBox asigne direcciones IP de forma automática. Un ejemplo de configuración válida se muestra en la Fig. 2.

Una vez creado el adaptador, estaremos en disposición de añadir la máquina virtual. Para ello, será necesario descomprimir la máquina virtual provista en el directorio utilizado por VirtualBox. Tras ello, basta con pulsar sobre *Machine > Add...* en la ventana principal de VirtualBox, y seleccionar el fichero con extensión `.vbox` correspondiente. Al aceptar, y si no hubiera ningún problema, debería aparecer la máquina virtual **PRACTICAS-ASI** en el listado de la interfaz principal.

La máquina virtual provista se proporciona con un primer adaptador de red configurado en modo *NAT*. Resulta necesario, por tanto, añadir un segundo adaptador en

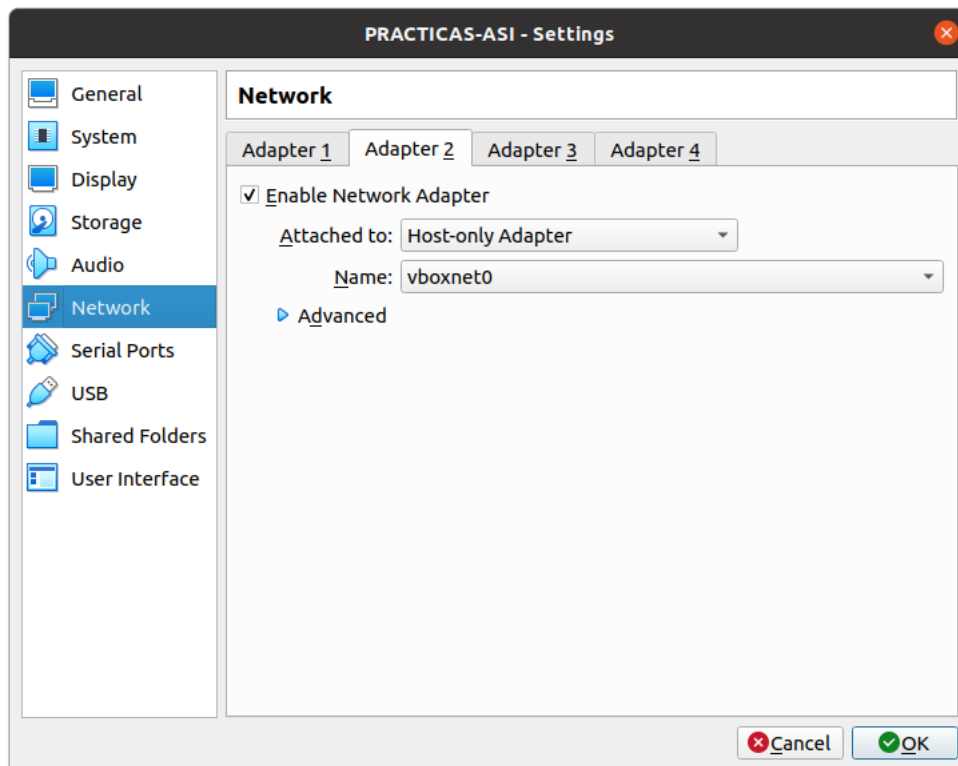


Figura 3: Parámetros relativos al segundo adaptador asociado a la máquina virtual.

modo *Host-Only Adapter* al que se le asignará el adaptador de la máquina anfitriona que acabamos de configurar. Para ello, basta con acceder a los parámetros de la máquina virtual (botón *Settings*), sección *Network*, y seguidamente a la pestaña correspondiente al segundo adaptador. Aquí, se debe marcar la opción *Enable Network Adapter* y cambiar el modo de red a *Host-Only Adapter*, asegurándonos que el nombre del adaptador del anfitrión asociado es el que hemos configurado previamente (por defecto, *vboxnet0*). Un ejemplo del resultado se muestra en la Fig. 3.

1.2. Arranque y configuración de la máquina virtual

Como se indicó previamente, la máquina virtual viene preparada con el sistema operativo Ubuntu 20.04 instalado. Contiene un usuario ya creado y añadido en la lista de *sudoers* con la siguiente información de acceso:

- Nombre de usuario: **ubuntu**
- Contraseña: **ubuntu**

En este punto, y antes de replicar la máquina virtual para simular el *cluster*, conviene arrancarla y comprobar que funciona correctamente. Cabe señalar que esta máquina

también tiene incorporadas las denominadas VirtualBox Guest Additions, versión 6.1.26, que proporcionan funcionalidad adicional al sistema invitado. En particular, algunas de las opciones que pueden resultar de mayor interés en la configuración de la máquina virtual son:

- *Compartición de portapapeles y transferencia de ficheros drag'n'drop.* La primera opción permite que el contenido que se copie en la máquina anfitriona esté disponible en el portapapeles del sistema invitado, y viceversa. Por su parte, la segunda opción permite pinchar y arrastrar ficheros entre ambos sistemas para transferirlos de manera más sencilla. Se habilitan en la sección *General*, pestaña *Advanced*.
- *Soporte a dispositivos USB 2.0 y 3.0.* Si se planea utilizar dispositivos USB para transferir ficheros de gran tamaño desde el sistema invitado, puede activarse esta opción en la sección *USB*.

Se recomienda que, en caso de haber activado alguna de estas funcionalidades o haber modificado algún parámetro de configuración, se compruebe que la máquina virtual funciona adecuadamente antes de clonarla. De igual modo, se recuerda que la entrega de los alumnos se probará sobre la máquina virtual provista inalterada, por lo que se requiere precaución a la hora de hacer cambios en la configuración de la máquina virtual que puedan afectar a la solución propuesta.

1.3. Clonación de máquinas virtuales

Una vez configurada la máquina virtual y comprobado que funciona correctamente, el último paso corresponde al replicado de la misma para disponer de tantas máquinas como nodos conformarán el *cluster*. Este proceso es sencillo, y sus pasos se enumeran a continuación:

1. Habiendo seleccionado la máquina virtual a clonar, pulsar sobre el menú *Machine > Clone...*
2. Sobre el cuadro de diálogo, deberemos otorgar un nombre a la nueva máquina virtual que se clonará a partir de la original. Adicionalmente, es importante marcar la opción *Generate new MAC addresses for all network adapters* para evitar conflictos en la red. El resto de opciones pueden permanecer desactivadas, tal como aparece en la Fig. 4.
3. En el siguiente paso, entre las opciones *Full clone* y *Linked clone*, elegir la que más se ajuste a las necesidades. Se recomienda *Full clone* por ser más sencilla, si bien supone un mayor espacio consumido en disco.

Una vez clonada la máquina, es necesario asegurarse de que los adaptadores de red de cada una de las copias están configurados en modo *NAT* y *Host-Only Adapter*, y que este último a su vez tiene asignado el adaptador de red del anfitrión adecuado.

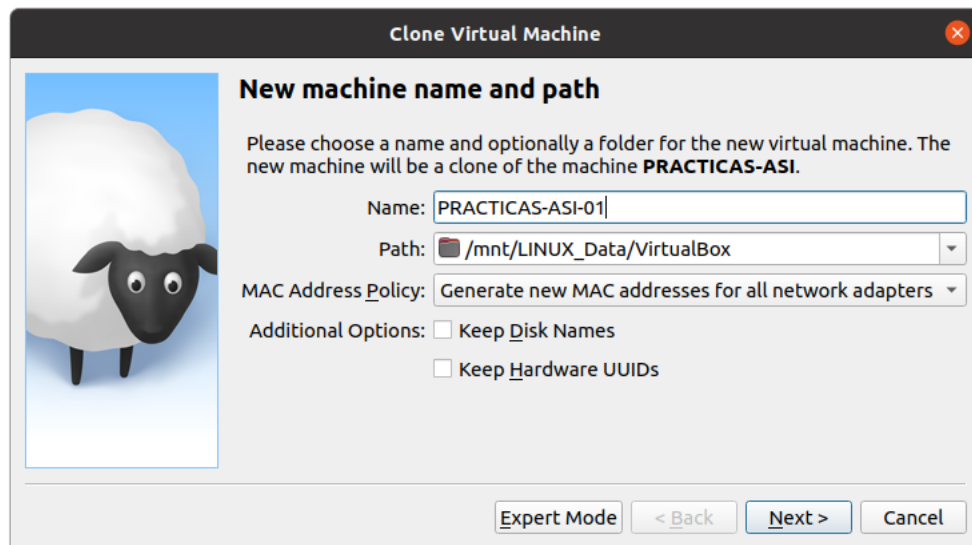


Figura 4: Opciones de clonado de máquinas virtuales.

1.4. Consideraciones generales

Una vez configurado el sistema, es necesario tener en cuenta una serie de consideraciones en la realización del proyecto:

- No debe hacerse ninguna suposición acerca de cuántas máquinas formarán parte del *cluster*. Durante la corrección del proyecto, podrá utilizarse un número arbitrario de máquinas. La solución implementada debe ser genérica para este parámetro. A pesar de lo anterior, y como orientación, se recomienda utilizar 3 máquinas: una desde la cual el administrador lanzará los mandatos, otra donde se instalará y configurará remotamente la parte cliente de los servicios, y una tercera destinada a instalar y configurar remotamente la parte servidora de los servicios.
- La máquina virtual se provee con únicamente un disco instalado desde el cual arranca el sistema operativo. Para la realización de este proyecto práctico, en particular para los servicios de montaje, RAID y LVM, puede resultar necesario añadir más unidades de almacenamiento a la máquina virtual que se pretenda configurar. Si bien este proceso es sencillo, se recoge más información al respecto en el Anexo B de este documento.
- No debe hacerse ninguna suposición acerca del nombre de las máquinas ni tampoco de sus direcciones IP (rango, máscara, etc.). Sí podrá asumirse, no obstante, que ya tienen configurados sus parámetros de red, es decir, que la solución implementada no deberá realizar ninguna configuración de red sobre las máquinas. Es posible encontrar más información en relación a la configuración de red de las máquinas virtuales en el Anexo B de este documento.

- La máquina provista tiene instalado y configurado el servidor SSH de tal modo que el superusuario puede acceder a otras copias de sí misma sin introducir contraseña. Esto permite que puedan realizarse operaciones en remoto de forma desatendida sin tener que escribir ningún tipo de credenciales a mano.

2. Desarrollo del proyecto práctico

El objetivo del proyecto práctico es programar un *script* en lenguaje *bash* que sea capaz de instalar y configurar de forma autónoma un conjunto de servicios en un grupo de máquinas interconectadas por una red local formando un *cluster*. El *script* deberá llamarse `configurar_cluster.sh` y su funcionamiento será el siguiente:

1. El administrador inicia sesión en cualquiera de las máquinas de la red.
2. El administrador prepara un fichero de configuración del *cluster*, indicando qué servicios deben ser instalados en cada máquina. El contenido de este fichero se explicará en detalle en la Sección 2.1.
3. El administrador ejecuta el *script* `configurar_cluster.sh`, pasándole el fichero de configuración del *cluster* como parámetro.
4. De forma autónoma, el *script* `configurar_cluster.sh` se conecta por SSH a cada una de las máquinas especificadas, instalando y configurando los servicios indicados.

Por tanto, el *script* principal a programar será invocado del siguiente modo:

```
# ./configurar_cluster.sh fichero_configuración
```

Una vez ejecutado este mandato, el *script* deberá ir informando por su salida estándar de las operaciones que realice. El formato de estos mensajes queda a elección de los alumnos, pero éstos deberán ser suficientemente explicativos para poder seguir las acciones del *script*. Los mensajes de error deberán imprimirse por la salida de error estándar. Se podrá optar por redirigir al dispositivo nulo la salida de aquellos mandatos cuya salida dificulte la lectura de la traza de progreso.

El *script* podrá invocar *scripts* auxiliares proporcionados por los alumnos. Obsérvese que es responsabilidad del *script* principal copiar los *scripts* auxiliares a las máquinas objetivo en caso de ser necesario. Resolver la práctica en uno o varios *scripts* queda completamente a la elección de los alumnos.

IMPORTANTE: todas las operaciones de administración deberán realizarse dentro de los *scripts* proporcionados, es decir, sin requerir operaciones manuales adicionales. Esto incluye, por ejemplo, la instalación de los paquetes necesarios. El objetivo es que la solución funcione correctamente ejecutada como superusuario sobre la máquina virtual **tal cual se proporciona**. Cualquier solución que no cumpla con este requisito será considerada incorrecta.

2.1. Fichero de configuración del cluster

El fichero de configuración del *cluster* será un fichero de texto plano. El *script* leerá el fichero y ejecutará cada una de las instrucciones de administración indicadas por orden de aparición. El formato del fichero de configuración del *cluster* será el siguiente:

- Las líneas en blanco serán ignoradas.
- Las líneas que comiencen por almohadilla (símbolo #) serán consideradas comentarios y, por tanto, ignoradas.
- Las líneas de configuración deberán cumplir el siguiente formato:

```
maquina-destino nombre-del-servicio fichero-de-perfil-de-servicio
```

donde *maquina-destino* es la dirección IP de la máquina donde se desea realizar la configuración, *nombre-del-servicio* es el nombre del servicio que se desea instalar y configurar, y *fichero-de-perfil-de-servicio* es la ruta, absoluta o relativa, hacia el fichero de configuración adicional con los parámetros del servicio que se desea configurar.

Cualquier línea que no cumpla este formato deberá ser detectada y considerada errónea. Se deberá imprimir un mensaje de error con la línea incorrecta por la salida de error estándar, interrumpir la ejecución y devolver un valor distinto de 0.

A continuación, se presenta un ejemplo que configura un RAID y un servidor NFS en la máquina 192.168.0.1, y un cliente NFS en el resto de máquinas que conforman el *cluster* de ejemplo:

```
# Máquina servidora
192.168.0.1 raid raid.conf
192.168.0.1 mount mount_raid.conf
192.168.0.1 nfs_server nfs_server.conf

# Máquinas cliente
192.168.0.2 nfs_client nfs_client.conf
192.168.0.3 nfs_client nfs_client.conf
192.168.0.4 nfs_client nfs_client.conf
192.168.0.5 nfs_client nfs_client.conf
```

Como se puede ver, esta configuración hace uso además de cuatro ficheros de perfil de servicio: *raid.conf*, *mount_raid.conf*, *nfs_server.conf* y *nfs_client.conf*. El formato de estos ficheros se describirá en la Sección 2.2. Asimismo, para facilitar el desarrollo del *script* por parte de los alumnos, el Apéndice A contiene algunos ejemplos de ficheros de entrada y sus correspondientes salidas, tanto cuando los ficheros de configuración son correctos como cuando contienen algún tipo de error.

2.2. Servicios requeridos

El *script* `configurar_cluster.sh` deberá ser capaz de instalar y configurar una serie de servicios en las máquinas que se soliciten. Cada servicio tendrá unos parámetros asociados, dependiendo de su funcionalidad. Como norma general, el código deberá comprobar que los parámetros de cada servicio son correctos, y mostrar qué errores se detectaron sin fallar, congelarse, dejar la máquina en un estado inutilizable, etc. En caso de detectarse algún error, sea de formato, sea de ejecución, el *script* debe devolver un código de error distinto de 0 y terminar la ejecución, pudiendo quedar servicios sin configurar. Además deberá realizar autónomamente las siguientes tareas cuando sea necesario:

- Instalación de software necesario para cada servicio si no se encuentra presente en la máquina.
- Creación de directorios necesarios, como puntos de montaje, etc.

A continuación se describe cada uno de los servicios y sus parámetros en detalle.

2.2.1. Montaje

Montaje de un dispositivo en un punto del árbol de directorios. Esto implica, además, que el sistema debe ser configurado para que el dispositivo se monte siempre en ese punto al arrancar la máquina (fichero `/etc/fstab`). El nombre del servicio es `mount`. El fichero de perfil de servicio debe ser un fichero de texto plano con 2 líneas. El formato de estas líneas debe ser:

```
nombre-del-dispositivo  
punto-de-montaje
```

El *script* debe crear el directorio escogido como punto de montaje si no existe. En caso de existir, deberá comprobar que el directorio está vacío y, si no lo estuviera, terminar la ejecución con un error. Un ejemplo del contenido del fichero de perfil de servicio sería:

```
/dev/sda2  
/home
```

2.2.2. RAID

Configuración de un RAID software, usando `mdadm`. El nombre del servicio es `raid`. El fichero de perfil de servicio debe ser un fichero de texto plano con 3 líneas. El formato de estas líneas debe ser:

```
nombre-del-nuevo-dispositivo-raid  
nivel-de-raid  
dispositivo-1 [dispositivo-2 ...]
```

Si alguno de los dispositivos utilizados en el RAID no existiera o si el nivel de RAID fuese incorrecto, bien por no representar un nivel RAID, bien por no corresponder al número de dispositivos provisto, la configuración del servicio debe ser abortada y devolver un error. Una vez configurado el RAID, la configuración debe ser persistente a través de los consiguientes reinicios de la máquina (fichero `/etc/mdadm/mdadm.conf` o equivalente según distribución). A continuación, se muestra un ejemplo de fichero de perfil de servicio en el que se configura un RAID nivel 1 usando los dispositivos `/dev/sdb1` y `/dev/sdc1`.

```
/dev/md0
1
/dev/sdb1 /dev/sdc1
```

2.2.3. LVM

Configuración de un grupo de volúmenes lógicos. El nombre del servicio es `lvm`. El fichero de perfil de servicio debe ser un fichero de texto plano con 3 o más líneas. El formato de estas líneas debe ser:

```
nombre-del-grupo-de-volúmenes
lista-de-dispositivos-en-el-grupo
nombre-del-primer-volumen tamaño-del-primer-volumen
nombre-del-segundo-volumen tamaño-del-segundo-volumen ...
```

El número de líneas de este fichero es variable, ya que depende de cuántos volúmenes lógicos se deseen crear. El *script* debe comprobar que no se excede el tamaño del grupo al crear los volúmenes, para lo cual se puede recurrir a los mandatos `vgdisplay` y `vgs`.

A continuación se muestra un ejemplo de fichero de perfil de servicio. En este ejemplo, se configura un grupo llamado `serverdata` usando los dispositivos `/dev/sdb1`, `/dev/sdc1` y `/dev/sdd1` y se crean dos volúmenes lógicos: `software`, de 100GB, y `users`, de 500GB.

```
serverdata
/dev/sdb1 /dev/sdc1 /dev/sdd1
software 100G
users 500G
```

2.2.4. Servidor NIS

Configuración de un servidor NIS. El nombre del servicio es `nis_server`. El fichero de perfil de servicio debe ser un fichero de texto plano con 1 línea. El formato de esta línea debe ser:

```
nombre-del-dominio-nis
```

Ejemplo de un fichero de perfil de servicio:

```
clusterASI2014
```

2.2.5. Cliente NIS

Configuración de un cliente NIS. El nombre del servicio es `nis_client`. El fichero de perfil de servicio debe ser un fichero de texto plano con 2 líneas. El formato de estas líneas debe ser:

```
nombre-del-dominio-nis
servidor-nis-al-que-se-desea-conectar
```

Ejemplo de un fichero de perfil de servicio:

```
clusterASI2014
192.168.0.1
```

2.2.6. Servidor NFS

Configuración de un servidor NFS con una serie de directorios a exportar. Dado que no se puede asumir ninguna configuración de red, los directorios deberán exportarse por todas las interfaces de red del servidor. El nombre del servicio es `nfs_server`. El fichero de perfil de servicio debe ser un fichero de texto plano con 1 o más líneas. El formato de estas líneas debe ser:

```
ruta-del-primer-directorio-exportado
ruta-del-segundo-directorio-exportado ...
```

El número de líneas de este fichero es variable, ya que depende de cuántos directorios se deseen exportar. Ejemplo de un fichero de perfil de servicio:

```
/home
/usr/local
```

2.2.7. Cliente NFS

Configuración de un cliente NFS. El nombre del servicio es `nfs_client`. El fichero de perfil de servicio debe ser un fichero de texto plano con una línea por sistema de ficheros que debe montarse. El formato de estas líneas debe ser:

```
ip-servidor ruta-de-directorio-remoto punto-de montaje
...
```

El número de líneas de este fichero es variable, ya que depende de cuántos directorios remotos se deseen montar. Cada uno de ellos deberá montarse con permisos de lectura y escritura para todos los usuarios. Además, el sistema debe ser configurado para que cada directorio remoto se monte siempre en ese punto al arrancar la máquina (fichero `/etc/fstab`). En caso de que un directorio exista previamente y no se encuentre vacío, la configuración del servicio deberá ser abortada y devolver un error. En caso de no existir el directorio del punto de montaje, este deberá ser creado. Ejemplo de un fichero de perfil de servicio:

```
192.168.0.1 /home /home
192.168.0.1 /usr/local /usr/local
```

2.2.8. Servidor de *backup*

Configuración de un servidor de *backup*. El nombre del servicio es **backup_server**. El fichero de perfil de servicio debe ser un fichero de texto plano con 1 línea. El formato de esta línea debe ser:

```
directorio-donde-se-realiza-el-backup
```

En caso de no existir el directorio donde se realizará la copia de seguridad, deberá crearse. En caso de existir y no encontrarse vacío, la configuración del servicio deberá ser abortada y devolver un código de error distinto de 0. Ejemplo de un fichero de perfil de servicio:

```
/storage/backup
```

2.2.9. Cliente de *backup*

Configuración de un cliente de *backup*. El nombre del servicio es **backup_client**. El fichero de perfil de servicio debe ser un fichero de texto plano con 4 líneas. El formato de estas líneas debe ser:

```
ruta-del-directorio-del-que-se-desea-hacer-backup
dirección-del-servidor-de-backup
ruta-de-directorio-destino-del-backup
periodicidad-del-backup-en-horas
```

El *script* debe configurar el *backup* para que se realice de forma periódica. En este ejemplo de fichero de perfil de servicio fijamos que se haga un *backup* del directorio **/home** cada 24 horas dentro del directorio **/storage/backup/node5** de la máquina 192.168.0.1:

```
/home
192.168.0.1
/storage/backup/node5
24
```

Nota sobre los servicios de *backup*: se deja a decisión de los alumnos la elección de la tecnología de *backup* empleada en la práctica. Se debe justificar esta elección en la memoria. Se recuerda a los alumnos que no todas las tecnologías posibles siguen un enfoque cliente-servidor (algunas emplean sólo un cliente, por ejemplo). El hecho de que los servicios especificados incluyan un cliente y un servidor no implica que la tecnología empleada tenga forzosamente que emplear este modelo. Se escoja la tecnología que se escoja, se debe poder configurar usando los mecanismos que se han indicado. Esto quiere decir, por ejemplo, que si se escoge una solución carente de parte servidora, la configuración de esta parte debe asegurarse, al menos, de que el directorio donde se realizará el *backup* exista y pueda ser escrito.

2.3. Memoria de la práctica

Los alumnos deberán elaborar una breve memoria del proyecto práctico, explicando el funcionamiento de la solución propuesta, así como cualquier dificultad encontrada u otros aspectos de interés. La memoria debe ser un documento formal, correctamente estructurado, redactado y formateado usando alguna herramienta de procesado y edición de textos (Microsoft Word, L^AT_EX, etc.) que permita su exportación a formato PDF. Debe contener figuras explicativas, tablas y otros recursos que ayuden a la comprensión del documento. **No se debe descuidar la redacción de la memoria**, ya que podría afectar negativamente a la nota final.

IMPORTANTE: la memoria **NO** debe contener el código de la solución implementada, el cual se entregará aparte.

3. Entrega

Los alumnos deberán entregar la práctica a través de Moodle en el apartado habilitado a tal efecto. La entrega deberá constar de **un único fichero comprimido** (en formato .zip, .rar, .tar.gz, etc.). El contenido de dicho fichero deberá ser:

- *Script configurar_cluster.sh*.
- Cualquier otro fichero auxiliar del que haga uso el *script* anterior.
- Memoria de la práctica en formato PDF.

4. Evaluación

La evaluación de la práctica se llevará a cabo usando un conjunto arbitrario de instancias de la misma máquina virtual puesta a disposición del alumno para su realización. La calificación otorgada será de 0 a 10 puntos, repartidos de la siguiente forma:

Parte	Puntuación máxima
Montaje	0.75 puntos
Configuración RAID	1.25 puntos
Configuración LVM	1.25 puntos
Configuración NIS	1.25 puntos
Configuración NFS	1.25 puntos
Configuración <i>backup</i>	1.25 puntos
Estructuración, legibilidad y mantenibilidad del código	1 punto
Calidad de la memoria	2 puntos

A. Ejemplos de entradas y salidas esperadas

En este apéndice se incluyen algunos ejemplos de ficheros de entrada que serán proporcionados al *script* a desarrollar y el tipo de salida que se espera obtener para dichas entradas en el proceso de corrección. Esta documentación **no pretende ser una lista exhaustiva** de todos los casos que serán probados, pero sí proporcionar una idea de cómo es la batería de pruebas empleada en la corrección.

A.1. Montaje (mount)

1. *Perfil de servicio recibido:*

```
/dev/sdb1  
/un/directorio/que/no/existe
```

Salida esperada: asumiendo que `/dev/sdb1` existe, no debe producirse ningún error y el valor devuelto debe ser 0. El directorio `/un/directorio/que/no/existe` ha sido creado y `/dev/sdb1` está montado sobre el mismo. Al reiniciar, `/dev/sdb1` se monta automáticamente.

2. *Perfil de servicio recibido:*

```
/este/dispositivo/no/existe  
/directorio
```

Salida esperada: error de montaje, el dispositivo no existe o no puede ser montado. El directorio de montaje puede haber sido creado. Código de error distinto de 0.

3. *Perfil de servicio recibido:*

```
/dev/sdb1
```

Salida esperada: error de sintaxis, falta el punto de montaje. Código de error distinto de 0.

A.2. RAID (raid)

1. *Perfil de servicio recibido:*

```
/dev/md0  
5  
/dev/sdb1 /dev/sdc1 /dev/sdd1
```

Salida esperada: se ha creado un dispositivo RAID nivel 5 llamado `/dev/md0` que debe aparecer con tal nombre en ese momento y en cualquiera de los posteriores reinicios de la máquina. El código de retorno debe ser 0.

2. *Perfil de servicio recibido:*

```
/dev/md0
17
/dev/sdb1 /dev/sdc1 /dev/sdd1
```

Salida esperada: nivel de RAID inválido. El valor devuelto debe ser distinto de 0.

3. *Perfil de servicio recibido:*

```
array1
0
/dev/sda /dev/sdc1 /dev/sdd1
```

Salida esperada: asumiendo que `/dev/sda` es un dispositivo que ya contiene un sistema de ficheros, el mandato fallará. Esta configuración es correcta en lo que respecta al nombre del dispositivo RAID que debe crearse. El valor devuelto debe ser distinto de 0.

A.3. LVM (lvm)

1. *Perfil de servicio recibido:*

```
grupo1
/dev/sdb /dev/sdc
vol1 2G
vol2 4G
```

Salida esperada: se ha creado un grupo LVM llamado `grupo1` que abarca los dispositivos `/dev/sdb` y `/dev/sdc`. Dentro de ese grupo se han creado los volúmenes lógicos `vol1`, de 2GB, y `vol2`, de 4GB. La información puede ser verificada con los mandatos `pvdisk`, `vgdisplay` y `lvdisplay`. El valor devuelto debe ser 0.

2. *Perfil de servicio recibido:*

```
grupo1
/dev/sdb /dev/sdc
```

Salida esperada: falta especificar al menos un volumen lógico que crear. El valor devuelto debe ser distinto de 0.

3. *Perfil de servicio recibido:*

```
grupo1
/dev/sdb /dev/sdc
vol1 4G
vol2 20000G
```

Salida esperada: el tamaño de `vol2` es mayor que el disponible en el grupo de dispositivos. El volumen `vol1` puede haber sido creado o no, del mismo modo que queda a discreción de los alumnos configurar el resto de volúmenes posteriores a aquel que genera un error. Independientemente de esto, el valor devuelto debe ser distinto de 0.

A.4. Cliente NIS (`nis_client`)

1. *Perfil de configuración recibido:*

```
dominio
192.168.200.1
```

Salida esperada: asumiendo que la dirección IP pertenece al servidor de dominio NIS, se configurará el cliente para poder usar dicho servidor para la autenticación. Por ejemplo, el superusuario será capaz de hacer `su usuario`, siendo `usuario` un nombre de usuario presente en el servidor. El *script* no tiene por qué detectar que el servidor de dominio existe y puede ser contactado. El valor de retorno debe ser igual a 0.

2. *Perfil de configuración recibido:*

```
dominio
esto.no.es.una.ip
```

Salida esperada: error de sintaxis y devolución de un valor distinto de 0.

3. *Perfil de configuración recibido:*

```
dominio
```

Salida esperada: error por configuración incompleta y devolución de un valor distinto de 0.

A.5. Servidor NFS (`nfs_server`)

1. *Perfil de configuración recibido:*

```
/directorio/valido
```

Salida esperada: el servidor NFS queda instalado y configurado para exportar en todas las subredes a las que está conectado. Debe ser posible montar cualquiera de los sistema de ficheros exportados desde cualquier máquina conectada a alguna de las subredes del servidor.

2. *Perfil de configuración recibido:*


```
/directorio/invalido  
/directorio/valido
```

Salida esperada: si un directorio no existe o no es un directorio, debe imprimirse un error y no añadirlo a `/etc/exports`. El servidor NFS puede quedar instalado o no. Queda a discreción de los alumnos decidir si aquellas entradas válidas, pero el *script* debe devolver un valor distinto de 0 en cualquier caso.

A.6. Cliente NFS (`nfs_client`)

1. *Perfil de configuración recibido:*

```
192.168.0.1 /export/home /home
```

Salida esperada: asumiendo que el *host* 192.168.0.1 está exportando `/export/home`, el directorio se montará en `/home` y será accesible por cualquier usuario (los permisos de acceso de cada fichero y directorio se corresponderán con los exportados por el servidor). El valor de retorno debe ser 0. Si se reiniciase la máquina cliente, los sistemas de ficheros deben ser montados al arranque.

2. *Perfil de configuración recibido:*

```
192.168.0.1 /home
```

Salida esperada: error por falta de parámetros de configuración. El valor de retorno debe ser distinto de 0.

3. *Perfil de configuración recibido:*

```
host.invalido /home /home  
192.168.0.1 /shared /shared
```

Salida esperada: error debido a que el primer parámetro no es un nombre de *host* existente y el directorio remoto no puede ser montado (un error similar es que el servidor no esté exportando el directorio solicitado). Queda a discreción del *script* detenerse en el primer error o continuar, pero el valor de retorno debe ser distinto de 0 en cualquiera de los casos.

A.7. Servidor de *backup* (`backup_server`)

1. *Perfil de configuración recibido:*

```
/ruta_a_directorio_existente/
```

Salida esperada: si el directorio no está vacío, error por intentar usar un directorio con contenido previo como almacén de *backup* y valor de retorno será distinto de 0. Si el directorio está vacío, el servidor de *backup* queda configurado correctamente.

2. *Perfil de configuración recibido:*

```
/ruta_a_directorio_inexistente/
```

Salida esperada: el directorio será creado y valor de retorno igual a 0.

A.8. Cliente de *backup* (backup_client)

1. *Perfil de configuración recibido:*

```
/ruta_a_directorio_local_existente/  
10.0.0.1  
/ruta_a_directorio_remoto_existente/  
24
```

Salida esperada: asumiendo que la dirección IP pertenece al servidor de *backup*, si el directorio del servidor puede ser escrito desde el cliente, este queda correctamente configurado. En caso contrario, error por falta de permisos de escritura en el directorio de *backup* y valor de retorno distinto de 0.

2. *Perfil de configuración recibido:*

```
/ruta_a_directorio_local_inexistente/  
10.0.0.1  
/ruta_a_directorio_remoto_existente/  
24
```

Salida esperada: error por directorio local inexistente y retorno distinto de 0.

3. *Perfil de configuración recibido:*

```
/ruta_a_directorio_local_existente/  
10.0.0.1  
/ruta_a_directorio_remoto_inexistente/  
24
```

Salida esperada: error por directorio remoto inexistente y retorno distinto de 0.

4. *Perfil de configuración recibido:*

```
/ruta_a_directorio_local_existente/  
10.0.0.355  
/ruta_a_directorio_remoto_existente/  
24
```

Salida esperada: error por IP del servidor de *backup* incorrecta y valor de retorno distinto de 0.

A.9. Errores en el fichero principal

1. *Perfil de configuración inexistente, no accesible o vacío.*

Salida esperada: impresión del error pertinente y devolver un valor distinto de 0.

2. *Perfil de configuración recibido:*

```
# Este fichero contiene errores de sintaxis

# La línea en blanco no es un error
192.168.0.1 raid raid.conf
# En la línea siguiente falta el fichero de configuración
192.168.0.1 mount
```

Salida esperada: falta el fichero de configuración del servicio `mount`. Debe imprimirse la línea errónea y devolver un valor distinto de 0. Los servicios previos a dicha línea pueden haber sido ejecutados o no.

3. *Perfil de configuración recibido:*

```
# Este fichero contiene un servicio que no existe
192.168.0.1 trocola trocola.conf
```

Salida esperada: el servicio solicitado no existe. El valor devuelto debe ser distinto de 0.

B. Configuración adicional de VirtualBox

Este anexo cubre algunas pequeñas indicaciones adicionales sobre la configuración de VirtualBox que son necesarias para la realización del proyecto práctico o que facilitan su desarrollo, y que no vienen recogidas en la Sección 1.

B.1. Configuración de medios de almacenamiento

La adición de discos nuevos a una máquina virtual se hace a través del cuadro de configuración de dicha máquina, en el menú *Machine > Settings...* En el diálogo de configuración, se accede a la sección *Storage*, y en el panel de la derecha se pincha en *Controller: SATA* y luego en el botón *Adds hard disk* para añadir un disco duro, tal como refleja la Figura 5.

A la hora de crear el nuevo disco duro, VirtualBox dará a escoger algunas opciones de configuración relativas al mismo. Se aconseja emplear las opciones por defecto del mismo, en particular utilizar el formato VDI, tamaño dinámico y una capacidad mínima de 8GB. Al tener un comportamiento dinámico, el espacio de dicho disco no se aloja en el anfitrión en el momento de creación, sino que se lleva a cabo cuando necesita crecer por necesidades de almacenamiento del sistema invitado.

Los discos añadidos aparecerán en el sistema invitado como `/dev/sdb`, `/dev/sdc`, `/dev/sdd...` Para el caso particular de los servicios de montaje y LVM, puede resultar de interés particionar los nuevos discos y crear sistemas de ficheros en ellos. Para lo primero, podemos recurrir al mandato `parted` del siguiente modo:

```
# parted --script /dev/sdb \  
    mklabel gpt \  
    mkpart part0 ext4 0% 2GiB \  
    mkpart part1 ext4 2GiB 4GiB \  
    mkpart part2 ext4 4GiB 100%
```

Mediante este mandato de ejemplo, crearemos una nueva tabla de particiones GPT en el dispositivo `/dev/sdb`, dos particiones de 2GB cada una y una tercera partición que ocupará el espacio restante del dispositivo. Por ejemplo, si el disco duro tuviera una capacidad de 10GB, esta tercera partición tendría un tamaño de 6GB. Si bien este mandato definiría las particiones de tipo EXT4, esto no crearía el sistema de ficheros en cada una de ellas. Para crear un sistema de ficheros tipo EXT4 en la primera partición de las tres creadas, el mandato a ejecutar sería:

```
# mkfs.ext4 /dev/sdb1
```

Para eliminar un disco duro basta con seleccionar el dispositivo que se quiere eliminar en el cuadro de configuración anteriormente mencionado y, bien pinchar con el botón derecho para seleccionar la opción de eliminar, o bien pinchar en el botón *Removes selected storage attachment* con un menos en la parte inferior.

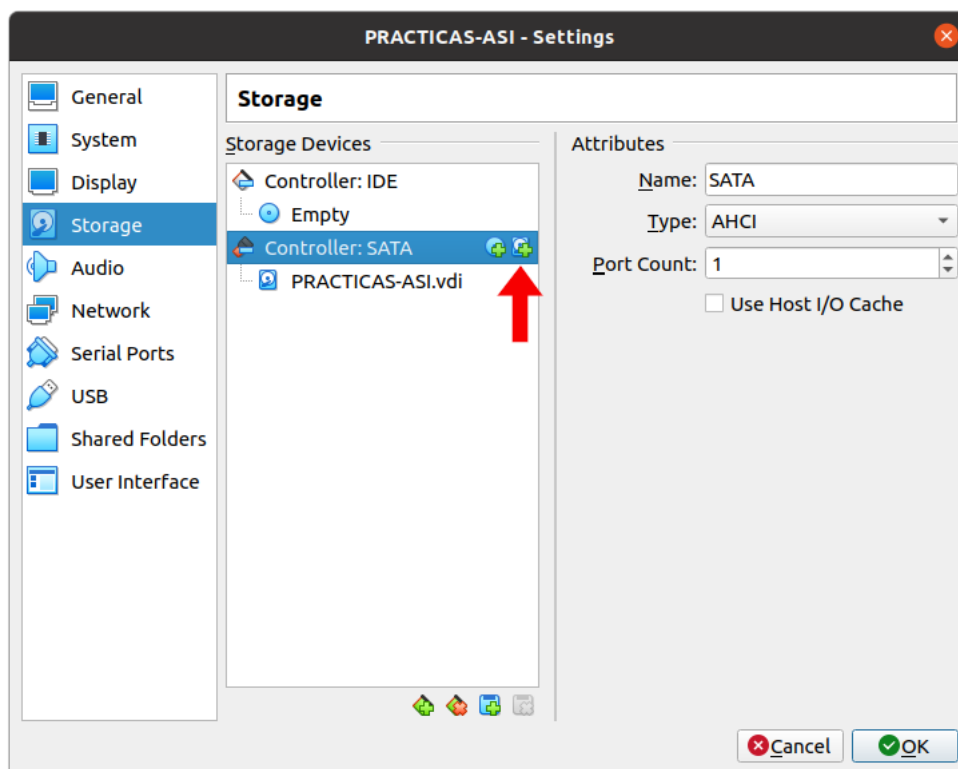


Figura 5: Configuración de dispositivos de almacenamiento de la máquina virtual.

B.2. Configuración de red adicional

Una vez configuradas las máquinas virtuales con el adaptador de red adicional en modo *Host-Only Adapter*, la interfaz correspondiente (`vboxnet0`) aparecerá en el anfitrión cuando cualquiera de ellas esté en ejecución. Dado que es una red de acceso directo, no es necesario configurar enrutamiento alguno. Por su parte, si se encuentra habilitado DHCP para dicho adaptador en VirtualBox, la máquina virtual recibirá una dirección IP dentro del rango establecido de forma automática. No obstante, dado que dicha dirección puede variar de un arranque a otro, puede resultar de interés asignar una dirección IP estática en el sistema invitado. En este caso, conviene recordar respetar los parámetros de red del adaptador (dirección IP y máscara), y además se aconseja asignar una IP fuera del rango destinado al servicio DHCP para evitar conflictos.

En caso de optar por la asignación por DHCP, existen varias formas de conocer la dirección IP asignada a cada máquina. Generalmente, la asignación se realiza de forma incremental comenzando por la primera dirección del rango, que por defecto es 192.168.56.3. No obstante, para confirmar con certeza cuál es la asignación, lo más sencillo es recurrir a estos dos métodos:

- Habiendo arrancado la máquina virtual, consultar los parámetros de red en el sistema invitado con el siguiente mandato:

```
$ ip address
```

- Desde el anfitrión, recurrir a las herramientas provistas por VirtualBox y VirtualBox Guest Additions para consultar los parámetros de red. Téngase en cuenta que en sistemas Mac podría ser necesario sustituir `vboxmanage` por `VBoxManage`, y en sistemas Windows por `VBoxManage.exe`.

```
$ vboxmanage guestproperty get nombre_de_maquina \
"/VirtualBox/GuestInfo/Net/X/V4/IP"
```

donde `nombre_de_maquina` es el nombre de la máquina virtual en ejecución, y `X` es el número del adaptador de red a consultar de dicha máquina.

En principio el tiempo de préstamo de las direcciones IP es suficientemente alto para asumir que, en combinación con la renovación, las direcciones otorgadas por el servidor DHCP no cambian.

B.3. Mandatos auxiliares

Esta sección recoge algunos mandatos que son útiles para gestionar las máquinas virtuales desde la terminal sin necesidad de usar la interfaz gráfica de VirtualBox.

- Arrancar una máquina virtual sin ventana gráfica:

```
$ vboxmanage startvm nombre_de_maquina --type headless
```

- Apagar una máquina virtual:

```
$ vboxmanage controlvm nombre_de_maquina poweroff
```

- Crear una instantánea de una máquina:

```
$ vboxmanage snapshot nombre_de_maquina take nombre_de_instantánea
```

Para que esto funcione, no debe existir ninguna instantánea con el mismo nombre.

- Restablecer el estado de una máquina al de una instantánea:

```
$ vboxmanage snapshot nombre_de_maquina restore nombre_de_instantánea
```

Si la máquina está arrancada, primero será necesario apagarla.

C. Virtualización mediante contenedores

Si bien la virtualización mediante máquinas virtuales a través de VirtualBox es la herramienta más sencilla para la realización del presente proyecto práctico, es cierto que requiere de elevados recursos físicos en la máquina anfitriona para poder ejecutar varias máquinas simultáneamente. Por este motivo, la máquina virtual provista tiene instalado Linux Container Daemon (LXD) preconfigurado para poder crear contenedores de Ubuntu 20.04 que se comportan de manera idéntica a dicha máquina virtual, pero empleando muchos menos recursos. Este anexo no pretende dar una descripción detallada sobre cómo funciona la virtualización mediante contenedores, sino proporcionar unas breves instrucciones acerca de cómo utilizarlos en el marco de este proyecto.

El servicio LXD tiene preconfigurado un espacio de almacenamiento para los contenedores y una red asociada, en particular la 10.0.5.0/24, por lo que, para el caso general, no es necesario realizar pasos adicionales de configuración. Además, se ha definido un perfil de LXD denominado **practicas-asi** que permite configurar fácilmente y de manera automática los contenedores creados. Este perfil configura el servicio SSH para que los contenedores puedan conectarse entre sí y desde el sistema anfitrión de manera idéntica a como se lleva a cabo entre diferentes copias de la máquina virtual provista, es decir sin necesidad de introducir credenciales como superusuario.

Así, el mandato para crear un contenedor de Ubuntu 20.04 denominado **node-01** utilizando el perfil mencionado anteriormente sería el mostrado a continuación. Es necesario tener en cuenta que la primera vez tardará unos minutos en descargar la imagen desde el servidor de imágenes, pero las siguientes ocasiones llevarán únicamente unos segundos:

```
$ lxc init ubuntu:20.04 node-01 -p practicas-asi
```

Los contenedores se encuentran preparados para recibir una dirección IP de forma automática mediante DHCP en su arranque. No obstante, podemos optar por asignarles una dirección IP estática en su lugar. Para ello, el mandato a lanzar para asignar al contenedor **node-01** la dirección IP 10.0.5.101 sería:

```
$ lxc config device override node-01 eth0 ipv4.address=10.0.5.101
```

En este punto, resultaría posible lanzar el contenedor creado con el mandato:

```
$ lxc start node-01
```

Para acceder y lanzar mandatos en los contenedores creados y en ejecución, existen dos alternativas principales:

- Por un lado, resultaría posible acceder mediante SSH como superusuario desde la máquina anfitriona al igual que entre máquinas virtuales.
- Como segunda alternativa, también resultaría posible abrir una terminal interactiva con el contenedor a través del mandato:

```
$ lxc exec node-01 /bin/bash
```

No obstante, habrá de tenerse en cuenta que esta alternativa no se considerará válida para lanzar mandatos de administración en la solución propuesta, puesto que se trata de simular el comportamiento de un cluster. Sólo se deberá usar a modo de depuración y testeo.

Finalmente, los mandatos para detener el contenedor y para eliminarlo serían los mostrados a continuación:

```
$ lxc stop node-01  
$ lxc delete node-01
```

IMPORTANTE: debido a las limitaciones existentes en LXD, a las restricciones de seguridad impuestas, y a que el sistema anfitrión y los contenedores comparten determinados recursos, el uso de contenedores no permite anexar dispositivos a los contenedores de la manera que requeriría este proyecto práctico. Por ello, no será posible llevar a cabo el testeo de los servicios de montaje, RAID y LVM mediante contenedores, siendo necesario emplear máquinas virtuales tal como se detalla en el resto del documento.