

PRACTICA 3:

CONFIGURACIÓN DE CONJUNTOS DE DISCOS REDUNDANTES (RAID) POR SOFTWARE EN LINUX

CONFIGURACIÓN DE UN ESCENARIO DE RED VIRTUAL PARA PRUEBA DE UN BALANCEADOR DE TRÁFICO

6 de noviembre de 2019

Objetivos

- Aplicar los conocimientos de protección de almacenamiento local mediante la configuración de un conjunto de discos en RAID.
- Comprender el funcionamiento de los mecanismos de interconexión de máquinas virtuales mediante bridges en Linux, mediante la configuración de un escenario virtual sencillo compuesto por dos redes locales interconectadas mediante un router.
- Comprender los algoritmos básicos de balanceo de carga entre servidores, mediante la configuración de un escenario sencillo compuesto por un balanceador que distribuye la carga entre tres servidores web.

Documentos y ficheros proporcionados

Se proporciona:

- Imagen de disco en formato qcow2 con sistema operativo Linux Ubuntu 18.04.
- Plantilla XML de definición de máquinas virtuales para libvirt.

Actividades a desarrollar

La práctica consiste en dos partes diferenciadas e independientes: parte A, dedicada a almacenamiento; y parte B, dedicada a configuración de escenarios y equipos de red.

Como resultado de la práctica se deberá entregar una memoria en la que se incluya la información y las respuestas solicitadas en las cajas de texto en negrita.

Antes de realizar cualquiera de las dos partes es necesario realizar las siguientes tareas previas:

- Cree un directorio nuevo en /mnt/tmp para almacenar todos los ficheros de la práctica:

```
cd /mnt/tmp
mkdir p3
cd p3
```

- Copie la imagen base de la máquina virtual a utilizar en esta práctica y descomprímala:

```
cp /mnt/vnx/repo/cdps/cdps-vm-base-p3.qcow2 .
```

La imagen proporcionada tiene registrados los usuarios cdps/cdps y root/cdps. Por seguridad, se recomienda entrar en las maquinas como usuario cdps y utilizar “sudo” para los comandos en los que se necesitan privilegios de superusuario (root).

- Copie además la plantilla de definición de VMs:

```
cp /mnt/vnx/repo/cdps/plantilla-vm-p3.xml .
```

Si necesita continuar la práctica en sesiones posteriores, no es necesario que comience desde cero la configuración de máquinas virtuales. Simplemente:

- Pare las máquinas virtuales ejecutando “**halt -p**” desde dentro de ellas.
- Copie el contenido del directorio /mnt/tmp/p3 a un pendrive o a su cuenta del laboratorio (ficheros *.xml y los ficheros de diferencias *.qcow2). Nota: para ahorrar espacio, tenga en cuenta que no es necesario copiar la imagen base de las maquinas virtuales (cdps-vm-base-p3.qcow2), ya que esta se puede volver a copiar del repositorio en cualquier momento.

Para restaurar la práctica simplemente copie los ficheros del pendrive o la cuenta al directorio /mnt/tmp/p3, copie de nuevo la imagen base y arranque las máquinas virtuales y sus consolas con los comandos descritos en el enunciado.

Parte A: Componentes de Almacenamiento

A.1 - Configuración de sistemas RAID locales por software

La herramienta *mdadm* de Linux permite gestionar metadispositivos (*md*) configurados sobre dispositivos (discos) físicos que se pueden configurar de acuerdo con determinadas configuraciones de RAID.

En el laboratorio simularemos este comportamiento mediante una máquina virtual basada en Ubuntu a la cual le presentaremos diferentes discos virtuales (que serán discos “físicos” para el sistema operativo ejecutando en la máquina virtual) y sobre ellos realizaremos las configuraciones RAID propuestas.

La actividad comenzará, por tanto, con la creación de la máquina virtual (MV) y los cinco discos de 512MB cada uno necesarios para realizar la práctica.

Para crear la máquina virtual:

- Haga una copia de la imagen descargada con nombre `s0.qcow2`:

```
cp cdps-vm-base-p3.qcow2 s0.qcow2
```

- Haga una copia de la plantilla XML para la nueva máquina virtual:

```
cp plantilla-vm-p3.xml s0.xml
```

- Edite la plantilla y sustituya los campos ‘XXX’ por los valores adecuados. Utilice el disco `s0.qcow2` como imagen y “`virbr0`” como bridge al que debe conectarse la MV. Posteriormente, arranque la máquina virtual con:

```
sudo virsh define s0.xml
sudo virsh start s0
```

Nota: el comando “`virsh create ...`” crea por defecto máquinas virtuales temporales, que se destruyen al apagarlas. Si queremos que las máquinas sean persistentes hay que utilizar “`virsh define`” y “`virsh start`”.

- Compruebe que la MV funciona adecuadamente accediendo a la consola mediante “`sudo virsh console s0`” y apunte la dirección IP asignada al interfaz `eth0` para poder acceder más cómodamente a la MV mediante `ssh`:

```
ssh cdps@<dir_ip>
```

- Apague la MV (p.e. con “`sudo halt -p`”).

Para crear y asignar los discos virtuales:

- Arranque el gestor de máquinas virtuales:

```
HOME=/mnt/tmp sudo virt-manager
```

- Acceda a la configuración de la MV (icono con la bombilla o la “i” en la consola de información que sale al hacer doble click en la MV) y cree (opción “+Agregar hardware”) 5 discos nuevos de 0,2 Gb, del tipo “Virtio disk”, con

nombre dX.img (siendo X=[1-5]) y localizados en el directorio de la práctica (/mnt/tmp/p3). Compruebe mediante ls que los ficheros dX.img se han creado en el directorio /mnt/tmp/p3.

- Arranque la MV de nuevo y, para facilitar copiar/pegar comandos, acceda a la consola textual mediante:

```
sudo virsh console s0
```

o, mucho mejor, mediante ssh a la dirección IP asignada a la MV:

```
ssh -X cdps@192.168.122.XXX
```

Una vez arrancada la máquina virtual esta deberá ser capaz de reconocer los cinco discos adicionales que se le han presentado (además del propio disco de sistema operativo). Para comprobarlo, podemos ejecutar el comando *fdisk -l* del sistema operativo.

Como ya se ha comentado, las configuraciones RAID se harán utilizando la herramienta *mdadm*; debemos comprobar que está instalada en el sistema operativo y si no, instalarla mediante el gestor de paquetes del sistema operativo.

Para poder utilizar los cinco discos presentados al sistema, primero hay que particionarlos. Podemos utilizar la herramienta gráfica *gparted* o el comando *fdisk*. Para cada disco definiremos una única partición (que cubra todo el disco) y la etiquetaremos del tipo *Linux RAID* –esto último no es estrictamente necesario, pero es útil asociar a cada partición un tipo.

Si se utiliza *fdisk*, para cada disco ejecute “sudo fdisk /dev/vdX” y:

- Cree una nueva tabla de particiones del tipo GPT (comando *g*).
- Cree una nueva partición con todo el contenido del disco (comando *n*).
- Cambie el tipo de la partición a Linux RAID (comando *t*).
- Escriba los cambios al disco (comando *w*).

Si se utiliza la aplicación *gparted* se debe formatear la partición como “ext3” y después elegir la opción “manage flags” para seleccionar el flag “raid”. Recuerde que para acceder a una aplicación gráfica se debe utilizar *slogin* con la opción “-X”, tal como se vio en la P1.

A continuación, comprobaremos que el módulo del kernel necesario para crear la configuración RAID5 que se utilizará en la práctica se puede cargar correctamente con el comando “*modprobe raid5*”. Para comprobar que el módulo está cargado utilizaremos:

```
lsmod |grep raid
```

El fichero */proc/mdstat* contiene información sobre las configuraciones RAID presentes en el sistema. Antes de comenzar, conviene cerciorarse de que no hay

ningún array/RAID siendo utilizado por el sistema (consúltese ese fichero mediante “cat /proc/mdstat”).

La configuración RAID propuesta es un RAID 5 para el que utilizaremos los cinco discos presentados al sistema operativo.

El comando *mdadm* necesario para crearlo es el siguiente:

```
sudo mdadm --create /dev/md0 --level=raid5 --raid-devices=5 /dev/vdb1 /dev/vdc1 /dev/vdd1 /dev/vde1 /dev/vdf1
```

donde

- */dev/md0* es el nombre del metadispositivo que utilizaremos para hacer referencia al grupo RAID
- *--level=raid5* indica el nivel de RAID que se va a configurar
- *--raid-devices=5* indica el número de discos de que constará el RAID
- */dev/vdbx* representa las particiones de los discos que se utilizarán para configurar el RAID

Si inmediatamente después de ejecutar el comando anterior se consulta el fichero */proc/mdstat* se podrá ver información sobre la creación del grupo RAID.

¿Qué contenido se muestra al leer este fichero? ¿Qué tamaño (neto) tendrá el grupo RAID recién creado?

Una vez creado el grupo RAID será necesario crear un sistema de ficheros sobre él, para que pueda ser utilizado por el sistema operativo. Esto se puede hacer con la instrucción *mkfs.ext4* (en realidad, podría utilizarse cualquier comando de la familia *mkfs* para crear diferentes tipos de sistemas de ficheros):

```
sudo mkfs.ext4 /dev/md0
```

Una vez tenemos un sistema de ficheros podemos montarlo en el sistema operativo:

```
sudo mount /dev/md0 /export 1
```

y comprobar el espacio disponible con *df -k*.

Pueden crearse directorios y ficheros sobre este sistema de ficheros para comprobar más tarde, durante las pruebas, que siguen siendo accesibles.

Si ahora ejecutamos la instrucción *fdisk -l* veremos que existe un nuevo dispositivo/disco.

¹ Si no existiera el directorio */export*, puede crearse con *mkdir /export*.

Puede consultarse el estado del grupo RAID creado con la instrucción:

```
mdadm --query /dev/md0
```

y pueden obtenerse más detalles con:

```
mdadm --detail /dev/md0
```

Obsérvese que, entre la información mostrada en la salida de este comando, se encuentra el estado del grupo RAID.

¿Qué estado presenta el grupo?

Igual que se puede consultar información sobre el grupo RAID, también se pueden hacer consultas sobre cualquiera de los componentes de ese grupo:

```
mdadm --query /dev/vdb1  
mdadm --examine /dev/vdb1
```

En este estado inicial el grupo RAID está limpio, en un estado OK.

Para poder comprobar constantemente el estado del raid, se recomienda abrir una nueva ventana en la máquina virtual y ejecutar:

```
watch mdadm --detail /dev/md0
```

Para poder comprobar el funcionamiento de la protección del grupo se puede simular el fallo de un disco y ver qué sucede. Esto puede hacerse con el comando:

```
mdadm --fail /dev/md0 /dev/vdb1
```

Si ahora vuelve a consultarse el estado del grupo RAID con `mdadm --query /dev/md0` aparentemente el grupo está bien; se ha perdido un disco, pero el grupo sigue operativo. Pero si comprobamos el estado real del grupo con `mdadm --detail /dev/md0`, se verá el nuevo estado del RAID.

El estado del grupo ha cambiado. ¿Cuál es?

Puede probarse la creación de ficheros debajo del directorio `/export` y ver qué sigue siendo accesible.

Igual que se ha simulado el fallo de un disco, puede simularse a continuación el reemplazo del disco fallido. Para ello, antes de nada, será necesario limpiar la información que del grupo RAID pueda tener la partición `/dev/vdb1` que es la que ha salido del grupo RAID.

```
mdadm --remove /dev/md0 /dev/vdb1  
mdadm --zero-superblock /dev/vdb1
```

Y puede volver a agregarse al grupo RAID como si se tratara de un disco nuevo:

```
mdadm --add /dev/md0 /dev/vdb1
```

¿Qué se observa ahora si se consulta el estado del disco (con `cat /proc/mdstat` o `mdadm --detail /dev/md0`)?

Puede comprobarse que durante este tiempo el contenido del directorio `/export` donde está montado el grupo RAID sigue siendo accesible.

Una vez recuperado el grupo RAID puede probarse qué sucede si se simula el fallo de dos discos.

¿Qué instrucciones podrían utilizarse para simular el fallo de dos discos? ¿Qué le ocurriría al grupo RAID?

Una vez un grupo RAID ha fallado (porque ha fallado más de un disco) no puede recuperarse. Es necesario pararlo y volverlo a crear si se considera necesario.

```
umount /export
mdadm --stop /dev/md0
```

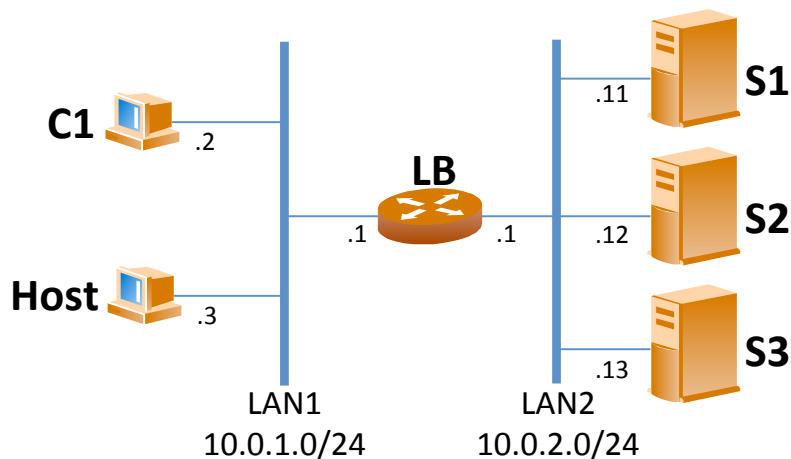
En lugar de recrear el grupo RAID 5 prueba a crear ahora un RAID10. Sabiendo que la palabra clave para identificar este tipo de configuración es `raid10` y que se desea utilizar cuatro discos,

¿Cuál sería el comando a utilizar?

Parte B: Componentes de Redes

B.1 - Escenario virtual de prueba de un balanceador de tráfico

En esta segunda parte de la práctica, se va a configurar un escenario de red pensado para evaluar y probar un paquete software con funcionalidades de balanceo de tráfico. En particular se utilizará *HAProxy* (ver manual en moodle) sobre el escenario de la figura, compuesto por dos redes y cinco máquinas virtuales (MV): tres servidores web (S1, S2 y S3), el balanceador (LB) y un cliente (C1). Adicionalmente se configurará un interfaz de red del host en la red LAN1, con el objeto de poder utilizar el navegador web u otras aplicaciones del host en el escenario.



B.2 - Creación del escenario de balanceo de carga

En primer lugar, se crearán las MVs y redes que forman el escenario. Cree los sistemas de ficheros COW que utilizará cada una de las MVs del escenario:

```
qemu-img create -f qcow2 -b cdps-vm-base-p3.qcow2 s1.qcow2
qemu-img create -f qcow2 -b cdps-vm-base-p3.qcow2 s2.qcow2
qemu-img create -f qcow2 -b cdps-vm-base-p3.qcow2 s3.qcow2
qemu-img create -f qcow2 -b cdps-vm-base-p3.qcow2 lb.qcow2
qemu-img create -f qcow2 -b cdps-vm-base-p3.qcow2 c1.qcow2
```

Cree el fichero XML de especificación de cada MV partiendo de la plantilla. Por ejemplo, para el servidor *s1*:

```
cp plantilla-vm-p3.xml s1.xml
vi s1.xml
```

Sustituya en la plantilla todos los campos marcados con XXX por los valores que correspondan en cada caso. Tenga en cuenta que:

- Los nombres de los bridges que soportan cada una de las LAN son LAN1 y LAN2

- La MV del balanceador (*lb*) debe tener dos interfaces de red, por lo que debe duplicar la sección `<interface>`.

Cree los bridges correspondientes a las dos redes virtuales:

```
sudo brctl addbr LAN1
sudo brctl addbr LAN2
sudo ifconfig LAN1 up
sudo ifconfig LAN2 up
```

Arranque el gestor de máquinas virtuales para monitorizar el arranque de las mismas:

```
HOME=/mnt/tmp sudo virt-manager
```

Arranque las máquinas virtuales utilizando el comando `virsh`. Por ejemplo, para *s1*:

```
sudo virsh define s1.xml
sudo virsh start s1
```

Acceda a las MV a través de la consola textual (en este caso no es posible acceder mediante `slogin`). Para ello, abra un nuevo terminal para cada MV y, por ejemplo, para *s1* ejecute:

```
sudo virsh console s1
```

Nota: es posible abrir un terminal nuevo y acceder a la consola con un solo comando, por ejemplo:

```
xterm -rv -sb -rightbar -fa monospace -fs 10 -title 's1' -e 'sudo
virsh console s1'
```

Nota: si se cambia el tamaño de la ventana en la que se accede a la consola de una máquina virtual, se debe ejecutar el comando `“resize”` para que se actualice el tamaño y los editores de texto como el `vi` funcionen correctamente.

Una vez arrancadas las máquinas virtuales, proceda a cambiarles el nombre modificando el fichero `/etc/hostname`. Para ello entre en la consola con usuario `cdps` y, por ejemplo, para *s1* ejecute:

```
sudo bash -c "echo s1 > /etc/hostname"
```

Edite, además, el fichero `/etc/hosts` y cambie la entrada asociada a la dirección `127.0.1.1` por el nombre de cada máquina. Por ejemplo, para *s1*:

```
127.0.1.1 s1
```

Rearranque cada máquina con `“reboot”` y tras rearrancar el `“prompt”` del sistema debería mostrar el cambio de nombre.

Proceda después a la configuración de red de cada una de las máquinas virtuales. Por ejemplo:

- Para *s1*:

```
sudo ifconfig eth0 10.0.2.11/24
sudo ip route add default via 10.0.2.1
```

- Para *lb*:

```
sudo ifconfig eth0 10.0.1.1/24
sudo ifconfig eth1 10.0.2.1/24
sudo bash -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
```

- Para el host:

```
sudo ifconfig LAN1 10.0.1.3/24
sudo ip route add 10.0.0.0/16 via 10.0.1.1
```

Nota: la configuración de red realizada con los comandos anteriores no es permanente. Si se rearrancan las máquinas se pierde. Es posible realizar una configuración permanente editando los ficheros `/etc/network/interfaces`, tal como se realizó en la P2. Para habilitar el reenvío (`ip_forward`) permanente en *lb* puede editar el fichero `/etc/sysctl.conf`

Finalmente, modifique las páginas web iniciales de los servidores *s1*, *s2* y *s3* para poder diferenciarlas (por defecto son todas iguales):

- En *s1*:

```
sudo bash -c "echo S1 > /var/www/html/index.html"
```

- En *s2*:

```
sudo bash -c "echo S2 > /var/www/html/index.html"
```

- En *s3*:

```
sudo bash -c "echo S3 > /var/www/html/index.html"
```

B.3 - Pruebas de conectividad y captura de tráfico

Una vez creado el escenario, compruebe que existe conectividad entre los sistemas del escenario. En particular:

- Que puede hacer ping desde el host y el *c1* hacia los servidores *s1*, *s2* y *s3*.
- Que desde un navegador del host puede acceder a las páginas principales de los servidores web de *s1* (`http://10.0.2.11`), *s2* (`http://10.0.2.12`) y *s3* (`http://10.0.2.13`). También puede comprobar el acceso a los servidores web mediante el comando `curl`, por ejemplo:

```
curl 10.0.2.11
```

Pruebe a capturar tráfico dentro de las MVs mediante:

```
sudo tcpdump -i eth0
```

O capturando en los interfaces LAN1 y LAN2 mediante wireshark desde el host (arránquelo desde el menú "Lab. Docentes DIT->Utilidades->Wireshark").

Analice como está realizada la conectividad mediante bridges y MVs. Para ello utilice los siguientes comandos:

```
brctl show
brctl show LAN2
```

```
brctl showmacs LAN2
```

Averigüe a que MV pertenece cada uno de los interfaces *vnet* que aparecen (mediante *showmacs* e *ifconfig* o capturando el tráfico con *tcpdump*).

Describa brevemente cómo se realiza la conectividad entre las máquinas virtuales, ilustrando la explicación con los resultados de los comandos “brctl show” e “ifconfig”. ¿Cómo se realiza la conexión del host al escenario virtual?

B.4 - Configuración del balanceador de carga

Como se ha mencionado, para esta práctica utilizaremos el software HAProxy que implementa un balanceador de carga que actúa como un proxy web. Esto es, el balanceador va a tener configurada una dirección IP virtual (vip), que es la dirección que conocen y a la que se conectan los clientes. Las peticiones web que recibe de éstos se van a redirigir hacia los servidores web según un algoritmo de distribución de carga configurable. Para ello establecerá nuevas conexiones entre el balanceador y los servidores.

Antes de realizar esta parte, lea la sección “Configure Load Balancing” del siguiente tutorial sobre HAProxy: <https://goo.gl/S6uz1i>

Para poner en marcha el balanceador:

- Pare el servidor web que esta corriendo en lb:

```
service apache2 stop
```

- Edite el fichero de configuración `/etc/haproxy/haproxy.cfg` y añada al final las siguientes líneas:

```
frontend lb
  bind *:80
  mode http
  default_backend webservers

backend webservers
  mode http
  balance roundrobin
  server s1 10.0.2.11:80 check
  server s2 10.0.2.12:80 check
  server s3 10.0.2.13:80 check
```

- Rearranque HAProxy mediante:

```
sudo service haproxy restart
```

A grandes rasgos, la configuración anterior pone al balanceador a escuchar en el puerto 80 de *lb*, define tres servidores activos (10.0.2.11:80, 10.0.2.12:80 y 10.0.2.13:80).

Pruebe a acceder desde el host o desde los clientes al URL `http://10.0.1.1` para comprobar a cuál de los servidores se está accediendo realmente. Para poder cargar a los servidores de peticiones y así poder ver el funcionamiento del balanceador en situaciones reales de carga, puede utilizar el siguiente comando que realiza peticiones continuas:

```
while true; do curl 10.0.1.1; sleep 0.1; done
```

Puede cambiar el valor del retardo en el comando “sleep 0.1” para modificar la carga de peticiones generadas (0.1=100 ms de retardo entre peticiones).

Configure el acceso a la página web de estadísticas de HAProxy siguiendo las indicaciones en <https://goo.gl/S6uz1i>. Configúrelas para que estén accesibles en el puerto 8001 y con usuario/clave admin/cdps.

Acceda a las estadísticas del balanceador (`http://10.0.1.1:8001`) y genere tráfico hacia los servidores para comprobar que se actualizan. Desactive alguno de los servidores (por ejemplo, deshabilitando su interfaz eth0 con “ifconfig eth0 down” para ver cómo las peticiones se redirigen a los que quedan activos. Vuelva a habilitar el interfaz eth0 y compruebe si el balanceador vuelve a enviar peticiones a ese servidor.

Pruebe algún algoritmo de distribución de carga alternativo al round-robin entre los que incluye HAProxy (por ejemplo, el round-robin con pesos) y describa las pruebas realizadas en la memoria, ilustrando la explicación con capturas de pantalla de la página de estadísticas o los resultados de comandos que considere adecuadas.