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

20 de octubre de 2015

Objetivos

- Aplicar los conocimientos de protección de almacenamiento local mediante la configuración de un conjunto de discos en RAID.
- Conocer los principios básicos de los sistemas de almacenamiento distribuidos mediante la configuración de un escenario basado en G*lusterfs*
- 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 14.04.
- Plantilla XML de definición de máquinas virtuales para libvirt.
- Manual del balanceador de tráfico Crossroads (disponible en moodle).

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.

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, por ejemplo de nombre p3:

```
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-vm-base-p3.qcow2.bz2 .
bunzip2 cdps-vm-base-p3.qcow2.bz2
```

  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/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"**.

- Copie el contenido del directorio /mnt/tmp/p3 a un pendrive (ficheros *.xml y *.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 al directorio /mnt/tmp/p3 y arranque de nuevo las máquinas virtuales y sus consolas con los comandos descritos anteriormente.

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 con diferentes máquinas virtuales basadas en Ubuntu a las cuales les 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 debe comenzar, por tanto, con la creación de los discos necesarios para realizar la práctica. Empezaremos creando cinco discos de 512 MB cada uno que presentaremos a cada máquina virtual (MV).

Para crear la maquina virtual:

- Cree el sistema de ficheros COW que utilizará la MV:
  ```
  qemu-img create -f qcow2 -b cdps-vm-base-p3.qcow2 s0.qcow2
  ```
- Arranque el gestor de máquinas virtuales:
  ```
  HOME=/mnt/tmp sudo virt-manager
  ```
- Cree una nueva máquina virtual de nombre s0, asociada a la imagen s0.qcow2, con 512M de memoria. En la pantalla final del asistente de creación, seleccione la opción "Personalizar configuración antes de instalar" y posteriormente:
  - Seleccione el formato "qcow2" para el disco principal (en opciones avanzadas de "Disk1").
  - Elimine la tarjeta de audio o configúrela como "ac97".
- Arranque la MV para comprobar que funciona adecuadamente y puede acceder a través de la consola.
- A continuación pare la máquina virtual mediante el comando "sudo halt -p" y una vez detenida, acceda a su configuración y cree 5 discos nuevos de 0,2 Gb, de tipo Virtio, con formato raw y localizados en el directorio de la práctica (/mnt/tmp/p3).
- Rearranque la máquina de nuevo y, para facilitar el copia/pega de comandos, acceda a la consola textual mediante:
  ```
  sudo virsh console s0
  ```

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 autodetect raid* (código *fd* de *fdisk*) –esto último no es estrictamente necesario, pero es útil asociar a cada partición un tipo.

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 vío en la P1.

A continuación, comprobaremos que los módulos de raid necesarios están cargados en el sistema operativo. Podemos comprobarlo con la instrucción *modprobe*.

```
root@ubuntu1:~# lsmod |grep raid
raid10                 48049  0
raid456                77529  1
async_raid6_recov      12955  1 raid456
async_memcpy           12650  1 raid456
async_pq               13154  1 raid456
async_xor              13019  2 async_pq,raid456
async_tx               13451 5 async_pq,raid456,async_xor…
raid6_pq               97812  2 async_pq,async_raid6_recov
raid1                  35528  0
raid0                  17772  0
```

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:

```
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):

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

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

```
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.

Puede consultarse el estado 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|examine} /dev/vdb1
```

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

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

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. Para comprobar el estado real del grupo, tiene que ejecutarse *mdadm –detail /dev/md0*.

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 inmediatamente después el fichero */proc/mdstat*?

Y si se utiliza el comando *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?

A.2 - Configuración de un sistema de ficheros distribuido con *Glusterfs* (Optativa)

A continuación se utilizarán dos máquinas virtuales para configurar un sistema de ficheros distribuido, es decir, un sistema de ficheros que puede ser utilizado por un cliente (a través de una red) pero que físicamente se encuentra distribuido entre varios sistemas.

Para ello, utilizaremos dos máquinas virtuales que tendrán configurado un sistema de ficheros en el directorio */export.* Por continuar con el punto anterior, pueden utilizarse dos máquinas virtuales con un sistema RAID5 o RAID10 configurado en ese directorio.

Para que esta solución funcione es necesario que ambas máquinas estén conectadas a través de la red.

Además, se podrá configurar una máquina virtual más (diferente de las dos anteriores) que actuaría como *cliente* del sistema de ficheros ofrecido por las dos primeras.

El sistema de ficheros que se instalará será *glusterfs*. Es necesario que las máquinas virtuales sobre las que se va configurar *glusterfs* tengan los paquetes necesarios instalados:

- *glusterfs-common*
- *glusterfs-server*
- *glusterfs-client*

Para comenzar la configuración, el primer paso es comprobar desde una de las máquinas la conectividad con la otra (*peer*).

```
root@ubuntu1:~# gluster peer probe ubuntu2
```

A continuación basta con ejecutar el siguiente comando para configurar un sistema de ficheros distribuido, en este caso con una configuración de replicación (*mirror*):

```
gluster volume create gv0 replica 2 ubuntu1:/export/brick
ubuntu2:/export/brick
```

Esta operación sólo es necesario ejecutarla en uno de los dos nodos; en el otro se ejecuta automáticamente.

Con *gluster volumen info gv0* se puede comprobar el estado del volumen creado.

¿Cuál es la salida?

Queda por arrancar el volumen para que pueda ser utilizado por los clientes.

```
gluster volumen start gv0
```

Se puede volver a comprobar el estado de gv0 *(gluster volumen info gv0*) para ver que este comando ha tenido efecto.

A partir de este momento, el volumen distribuido y replicado está listo para ser utilizado. Desde una de las propias máquinas servidores podemos montarlo:

```
root@ubuntu1:# mount -t glusterfs -o log-level=WARNING,log-
file=/var/log/gluster.log ubuntu1:/export/brick /mnt
```

Intente montarlo en una máquina independiente de las anteriores;

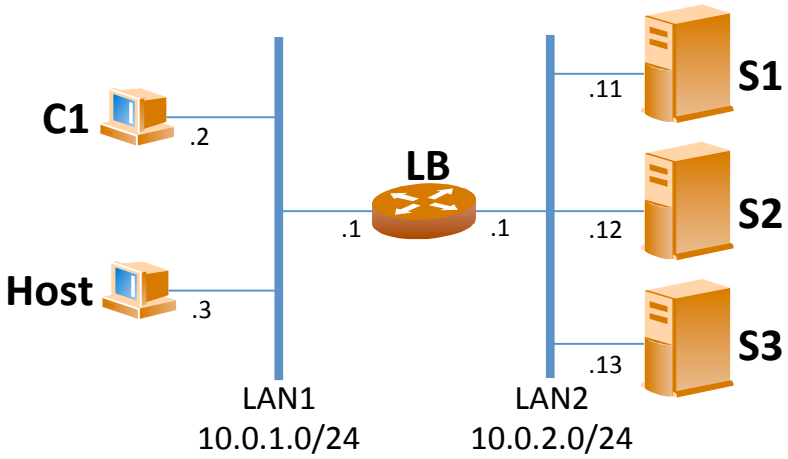
¿Qué paquetes relacionados con el sistema *gluster* serían necesarios?

Puede comprobarse que cuando se crea un fihero en */mnt* este aparece replicado en las dos máquinas que constituyen el cluster de *glusterfs* debajo de */export/brick*.

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 Crossroads (http://crossroads.e-tunity.com/) 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
gedit 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 maquinas virtuales utilizando el comando virsh. Por ejemplo, para *s1*:

```
sudo virsh create s1.xml
```

Para acceder a las consolas de las MVs existen dos opciones:

- Acceder a las consola gráfica a través de virt-manager, simplemente haciendo doble-click en la línea de cada MV. La desventaja de esta opción es que no permite realizar corta/pega de comandos de texto.
- Acceder a las consolas textuales (recomendada). 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'
  ```

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 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 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 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* (buscar en Internet como realizarlo).

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

- En *s1*:

```
echo S1 > /var/www/html/index.html
```

- En *s2*:

```
echo S2 > /var/www/html/index.html
```

- En *s3*:

```
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 ifconfigs 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 balanceador de carga

Como se ha mencionado, para esta práctica utilizaremos el software Crossroads 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.

Para poner en marcha el balanceador:

- pare el servidor web que esta corriendo en *lb*:
  ```
  service apache2 stop
  ```
- arranque el servidor de balanceo de carga mediante el comando:

```
xr --verbose --server tcp:0:80 --backend 10.0.2.11:80 --backend 10.0.2.12:80 --backend 10.0.2.13:80 --web-interface 0:8001
```

A grandes rasgos, el comando anterior pone al balanceador a escuchar en el puerto 80 de *lb* ("--server tcp:0:80"), define tres servidores activos ("--backend 10.0.2.11:80 --backend ...") y arranca un servidor web para la gestión del balanceador en el puerto 8001 de *lb*. Consulte la documentación de Crossroads para mas detalles sobre opciones de configuración (copia del manual en moodle).

Pruebe a acceder desde el host o desde los clientes al URL http://10.0.1.1 para comprobar a cual de los servidores se esta 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).

Acceda a la gestión web del balanceador (http://10.0.1.1:8001) y desactive alguno de los servidores para ver como las peticiones se redirigen a los que quedan activos.

Pruebe con al menos dos de los distintos algoritmos de distribución de carga que soporta xr y describa las pruebas realizadas en la memoria, ilustrando la explicación con las capturas de pantalla o resultados de comandos que considere adecuadas.

B.5 – Balanceo en función de carga de servidores (Opcional)

La opción "-be" del commando *xr* permite incorporar algoritmos de balanceo de carga externos, implementados mediante scripts. Cada vez que llega una conexión, el comando *xr* ejecuta un programa o script externo que decidirá a cual de los servidores enviar la conexión.

En esta parte opcional se propone escribir un script que envié las conexiones nuevas siempre al servidor menos cargado. Para ello se propone:

- Crear un script en python que periódicamente (p.e. cada 10 segs.) y mediante SSH consulte la carga de cada uno de los servidores y la almacene en un fichero (p.e. /tmp/server-loads.txt). El comando especifico para consultar la carga de un servidor podría ser:

  ```
  ssh 10.0.2.11 uptime
  ```

  Nota: busque en Internet información sobre cómo evitar que se solicite la clave de acceso al usar ssh (busque por ejemplo "ssh no password")

- Cree otro script siguiendo las indicaciones que aparecen en la página 10 del manual de Crossroads que consultando el fichero /tmp/server-loads.txt elija el servidor menos cargado.