

UNIVERSIDAD DE GRANADA

ISE

2018

Cuaderno de prácticas de ingeniería de servidores

Autor:

Francisco Navarro Morales

Índice

1. Práctica 1	2
1.1. Primera sesión	2
1.1.1. Introducción a los servidores.	2
1.1.2. Configuración de ubuntu server.	2
1.1.3. Particionado para ubuntu server	3
1.2. Segunda sesión.	5
1.3. Tercera sesión.	8

1. Práctica 1

1.1. Primera sesión

1.1.1. Introducción a los servidores.

1.1.2. Configuración de ubuntu server.

A continuación describiré el proceso para configurar ubuntu server en una máquina virtual (VirtualBox).

La primera pregunta que nos haremos será cuanta memoria RAM asignaremos a la máquina virtual. Ubuntu server podría funcionar con tan solo 500mb (sería una cantidad mínima a asignar) y, en caso de tratarse de un servidor que fuera a tener mucha carga, le podríamos asignar varios gigabytes. En mi caso voy a dejar la opción por defecto (1024 mb) ya que no sé que carga va a tener a priori la máquina y tampoco necesito optimizar los recursos hasta el punto de ahorrar en memoria RAM.

El tipo de disco también seleccionaré la opción por defecto de virtualbox.

Respecto al tipo de almacenamiento (**dinámico vs estático**) tenemos que tener en cuenta que el almacenamiento dinámico ofrece más versatilidad pero también produce una mayor fragmentación. Es decir, utilizar un tamaño fijo nos hará ocupar recursos innecesariamente desde un principio pero dará mejor rendimiento (sería una manera de optimizar en caso de saber de manera más o menos exacta las necesidades de memoria del servidor). En este caso utilizaremos reserva dinámica porque la fragmentación no es algo que nos deba preocupar para el desarrollo de las practicas.

Una vez creada la máquina virtual, vamos a añadir el ISO de instalación de ubuntu server y además vamos a añadir un segundo disco puesto que queremos utilizar tecnología RAID (redundant array independent disc). En concreto vamos a hacer un raid de tipo 1, que consiste en hacer un mirror (una copia exacta) de los datos en dos o más discos a la vez. Este tipo de configuración mejora los tiempos de lectura (aunque puede suponer empeorar el tiempo de escritura) y aporta mayor seguridad a la información almacenada (si un disco falla, tenemos el resto).

Para configurar los discos, click derecho en la máquina virtual, settings¿storage, en el controlador IDE añadiremos el ISO con ubuntu server y en los controladores SATA crearemos un nuevo disco (click en controller: SATA ¿botón cuadrado con

un +)

Iniciamos la máquina virtual, idioma inglés, install ubuntu server, seleccionamos el idioma, la configuración de teclado en español y rellenamos algunos valores como el nombre del host o del usuario principal.

Cuando nos pregunten si queremos encriptar el directorio home le decimos que no, puesto que lo vamos a encriptar manualmente junto con el resto del sistema.

1.1.3. Particionado para ubuntu server

El procedimiento es el siguiente:

- creamos la tabla de particiones para cada uno de los discos
- creamos el MD (multiple device-drive aka Linux software RAID)
- creamos el logic volume y las particiones de este
- creamos los volúmenes de cifrado
- formateamos y asignamos puntos de montaje a los volúmenes cifrados.

En el momento de realizar el particionamiento seleccionamos ‘Particionado manual’ y ahora seleccionamos cada uno de los discos y le decimos al asistente que cree una tabla de particiones en él.

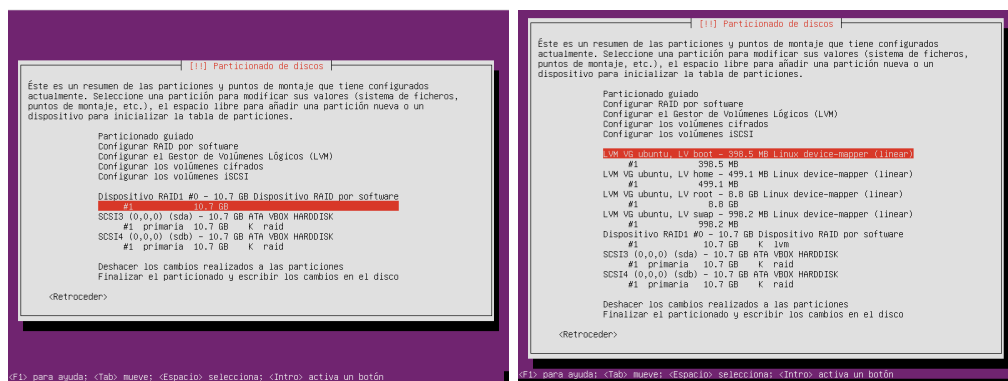


Figura 1: Visualización del esquema de particionado justo después de crear el volumen físico, md0, y justo después de configurar el LVM y crear las particiones.

Una vez creadas las particiones procederemos a crear un nuevo dispositivo md (raid por software). Se nos preguntará qué queremos hacer, le indicamos que crear un nuevo dispositivo MD, seleccionamos raid tipo 1 y le especificamos que utilizará dos discos, seleccionamos los dos discos con espacio y pulsamos enter para proceder. Finalmente pulsamos en Terminar. (ver figura 1).

Ahora tenemos definido un volumen físicos (PV, physical volumen) y en él, definiremos 4 volúmenes lógicos (LV, logical volume); root, boot, home y swap.

Para ello, crearemos el LVM (logical volume manager). Vamos a configuración de LVM, crear grupo de volúmenes (lo llamaremos ubuntu) y seleccionamos el dispositivo físico md0.

Ahora el procedimiento es el siguiente: dado que cifraremos los volúmenes lógicos por separado (más seguro y versátil que cifrar todo el LVM) y que, al cifrar un volumen lógico se pierde la información de este referente a aspectos como el formato o el punto de montaje...

Por tanto, una vez creado el LVM, seleccionaremos "crear volumen lógico" para crear cada nueva partición.

El orden no importa, pero dejaremos el root para el final para asegurarnos de asignarle toda la memoria posible. Respecto a cuanta memoria asignar: para boot bastaría con unos 200 mb, pero como tampoco andamos mal de memoria podemos asignar hasta unos 400 mb... En un pc asignaríamos bastante más memoria al home pero al tratarse de un servidor... no tendremos muchos datos en esta partición así que 500 mb está bien. Para swap podemos utilizar un gb aunque también estaría bien un poco menos (o un poco más pero no hay que hacerla demasiado grande). Respecto al formato... (que no deberíamos aplicar todavía puesto que de hacerlo se perdería al cifrar las particiones) swap será memoria de intercambio y las otras tres pueden ser ext4 o ext2, preferimos ext4 porque es más moderna y en teoría mejor. Una posible excepción sería el boot al que le pondremos ext2 (si queremos) porque las ventajas que aporta ext4 no se van a notar y ext2 es más 'ligero'. Como ya se ha especificado, al crear la partición para root (la última) asignamos todo el espacio restante (el que aparece por defecto). En la figura 1 se puede visualizar como debe quedar el esquema hasta el momento.

A continuación seleccionaríamos 'configurar los volúmenes de cifrado' y le damos a crear volúmenes de cifrado. Seleccionamos todas las particiones excepto el boot (se podría cifrar pero es más complicado). Las dejamos como están (opciones

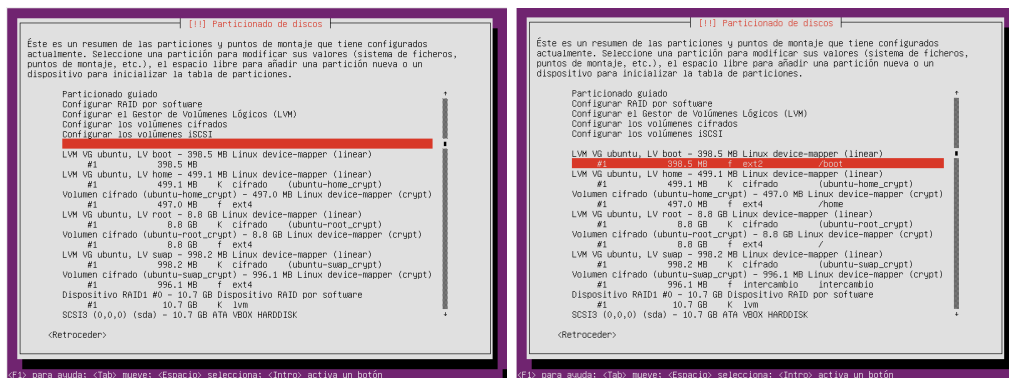


Figura 2: Visualización del esquema de particionado justo después de cifrar el sistema de archivos y justo después de asignar a las particiones formato y puntos de montaje.

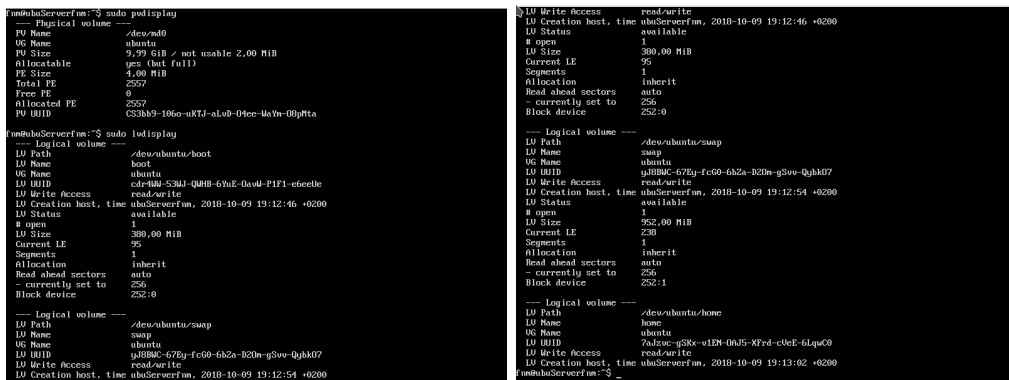


Figura 3: Comprobaciones post-instalación con pvdisplay y lvdisplay.

por defecto) y procedemos activando los volúmenes cifrados.

Debemos configurar las distintas particiones hasta pasar del esquema de la figura 3 al de la figura 3

Una vez esté todo como en la figura 3 podremos proceder pulsando ‘Finalizar el particionado y escribir los cambios en el disco’.

1.2. Segunda sesión.

En esta parte trabajaremos con una imagen de CentOS con el siguiente escenario: un cliente nos pide alojamiento para una web de vídeos y, debido a esto, /var se nos queda sin espacio. Entonces compramos un nuevo disco (sdb, en la práctica

será un disco virtual, simulado) y queremos mover `/var` a un nuevo volumen lógico con mayor tamaño.

La dinámica será la siguiente:

- creamos un nuevo volumen físico
- extendemos el LVM para añadir el nuevo disco.
- creamos una nueva partición o volumen lógico que será el nuevo `/var`
- creamos el sistema de archivos en la nueva partición
- montamos la partición en un nuevo punto de montaje
- copiamos toda la información del `/var` previo
- asignamos el punto de montaje `/var` a la nueva partición
- liberamos el espacio del antiguo `/var`

Para crear el volumen físico utilizaremos **`pvccreate /dev/sdb`** y comprobamos que se haya creado correctamente con **`pvs`**. ¿Haría falta crear la tabla de partición en el dispositivo? NO, no es necesario, podemos crearlo sin más.

A continuación extenderíamos el LVM con **`vgextend cl /dev/sdb`** (`cl` es el nombre que le da CentOS al LVM principal) y comprobamos que todo está bien con `vgs`.

Creamos un nuevo volumen lógico con **`lvcreate -L 4G -n newvar cl`** y comprobamos que todo está correcto con **`lvdisplay`**. Los volúmenes lógicos tienen la pinta **`/dev/cl/newvar`** o bien **`/dev/mapper/vgn-lua`**.

Ahora creamos el sistema de archivos con **`mkfs -t ext4 /dev/cl/newvar`**, aquí podríamos valorar algunas opciones como por ejemplo cambiar el tamaño de bloque para obtener un mejor rendimiento en ciertos casos.

Creamos un punto de mntaje y montamos el volumen lógico: **`mkdir /media/newvar && mount /dev/cl/newwvar /media/newvar`**

A continuación copiamos el var antiguo con **`cp -a /var/. /media/newvar`**. Un detalle importante es que deberíamos asegurar la atomicidad al copiar. Si copiamos cantidades enormes de datos de golpe podrían darse problemas por culpa del uso del servidor a la vez que se copian los datos. Una solución sería cortar el acceso al

```

[root@localhost fnm1# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0    8G  0 disk
└─sda1       8:1    0   400M  0 part /boot
└─sda2       8:2    0    7.6G  0 part
   └─cl-root 253:0    0    6.7G  0 lvm  /
   └─cl-swap 253:1    0    952M  0 lvm  [SWAP]
sdb          8:16   0    8G  0 disk
└─cl-newvar 253:2    0    4G  0 lvm  /var
sr0         11:0    1 1024M  0 rom
[root@localhost fnm1#

```

Figura 4: Output de lsblk tras completar todos los pasos

servidor mientras se hace la copia o (preferiblemente) cambiar el nivel de ejecución (targets de systemd) para asegurar que la copia se realiza con preferencia sobre cualquier otra acción: **systemctl isolate runlevel1.target**

A continuación tendríamos que editar el archivo **etc/fstab** y añadir la línea **‘/dev/mapper/cl-newvar /var ext4 defaults 00’**

Hacemos los cambios efectivos con **mount -a**, comprobamos que todo es correcto y, finalmente, procedemos a liberar el espacio del antiguo /var

Para ello tenemos que volver a la configuración original desmontando el nuevo /var (**umount /dev/mapper/cl-var**), esto hará que se vuelva a montar el antiguo var en /var. **lsblk** para comprobarlo. A continuación **mv /var /media/varold** (por si acaso), **mkdir /var**, **restorecon /var** y finalmente **mount -a**

Y ya estaría.

1.3. Tercera sesión.

En esta última sesión de la primera práctica vamos a hacer un RAID 1 para el var y encriptarlo.

Puedes ir usando `lsblk` para comprobar el esquema de dispositivos del sistema.

Para crear el RAID utilizaremos el comando `mdadm` (no viene instalado por defecto). Al abrir la máquina virtual no tendremos internet... hay que levantar la interfaz de red. Buscamos el nombre de la interfaz con `ip addr` y `ifup *interfaz*` para levantar la interfaz.

Comprobamos que tenemos internet con un ping y usamos yum para instalar `mdadm` **yum install mdadm**.

ahora el comando se usa tal que: **mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sdX /dev/sdY** (los dispositivos los hemos comprobado con `lsblk`)

ahora creamos el volumen físico en el nuevo raid: **pvcreate /dev/md0**

En este caso no nos vale con extender el viejo grupo de volúmenes... necesitamos uno nuevo para evitar que se siga escribiendo en el antiguo var hasta llenarlo.

vgcreate pmraid1 /dev/md0 (comprobamos con `vgs`).

Ahora sí, creamos el volumen lógico **lvcreate -L 1G -n newvar pmraid1** y comprobamos con `lvdisplay`.

mkfs -t ext4 /dev/mapper/pmraid1-newvar

mkdir /media/newvar && cp -a /var/. /media/newvar

añadimos al `/etc/fstab` :

/dev/mapper/pmraid1-newvar /var ext4 defaults 0 0

y procedemos como en la sesión anterior:

Hacemos los cambios efectivos con **mount -a**, comprobamos que todo es correcto y, finalmente, procedemos a liberar el espacio del antiguo `/var`

Para ello tenemos que volver a la configuración original desmontando el nuevo `/var` (**umount /dev/mapper/pmraid1-newvar**), esto hará que se vuelva a montar el antiguo var en `/var`. **lsblk** para comprobarlo. A continuación **mv /var /media/varold** (por si acaso), **mkdir /var**, **restorecon /var** y finalmente **mount -a**

Y ya estaría.

Finalmente nos tocaría cifrar el nuevo var. Instalamos la utilidad `cryptsetup`

con yum install... y, como al cifrar se borra todo... primero vamos a hacer una copia **mkdir /media/varRAID/ && cp -a /var/. /media/varRAID**

antes de formatear hay que desmontar el dispositivo: **umount /dev/mapper/pmraid1-newvar** pero es posible que no nos deje porque haya alguien utilizándolo... usando lsof (hay que instalarlo) en /var veríamos que dhcclient está escribiendo en var... podemos parar el proceso de dhclient con un kill -9 y su pid y luego probar a desmontar de nuevo.

Una vez desmontado... usamos **cryptsetup luksFormat /dev/mapper/pmraid1-newvar**

Como al encriptar se pierde el sistema de archivos, tendremos que crearlo otra vez. Pero primero hay que ‘abrir’ el sistema cifrado (algo parecido a montar un volumen lógico) con **cryptsetup luksOpen /dev/mapper/pmraid1-newvar pmraid1-newvar_crypt**

Aquí sería interesante usar el comando **shred** para meter basura en el volumen antes de cifrarlo para evitar posibles intentos de descifrar el volumen.

mkfs -t ext4 /dev/mapper/pmraid1-newvar_crypt

mkdir /media/newvar_crypt && mount /dev/mapper/pmraid1-newvar_crypt /media/newvar_crypt

cp -a /media/varRAID/. /media/newvar_crypt

editamos el fstab

/dev/mapper/pmraid1-newvar_crypt /var ext4 defaults 0 0

y el /etc/crypttab, para el cual necesitamos el UUID del dispositivo... podemos utilizar blkid:

blkid | grep crypto >> /etc/crypttab

y luego editar con vim hasta que quede algo así:

pmraid1-newvar_crypt /dev/md0 UUID=793427491729387129873273 none