

MEMORIA P5

#870984, Simon Gayan, Daniel, M, 3, A

#872815, Solana Melero, Carlos, M, 3, A

Configuración Inicial

Antes de iniciar las máquinas virtuales, hemos añadido un nuevo disco de 2GB en el campo de almacenamiento de cada máquina, utilizando el controlador SATA, este es el disco a particionar.

Hemos usado estos comandos

```
sudo parted /dev/sdb mklabel gpt —para crear la tabla de particiones
```

```
sudo parted /dev/sdb mkpart p1 ext3 0MB 1000MB —creamos la primera partición del 0 a 1000mb
```

```
sudo parted /dev/sdb mkpart p2 ext4 1000MB 2000MB —creamos la segunda partición del 1000 a 2000mb
```

```
sudo parted /dev/sdb align-check optimal 1 —vemos si cada una está alineada
```

```
sudo parted /dev/sdb align-check optimal 2
```

Tras crear las particiones, las hemos formateado y hemos montado :

```
sudo mkfs -t ext3 /dev/sdb1 — formateamos las particiones con el sistema de archivos que le corresponde
```

```
sudo mkfs -t ext4 /dev/sdb2
```

```
sudo mount -t ext3 /dev/sdb1 /media1 — montamos la partición en el directorio media1
```

```
sudo mount -t ext4 /dev/sdb2 /media2 — montamos la otra partición en otro directorio diferente para evitar errores
```

Configuración de Montaje Automático y Verificación

Para que se monten de manera automática hemos modificado así el fichero de fstab

```
/dev/sdb1 /media1 ext3 rw,relatime 0 0
```

```
/dev/sdb2 /media2 ext4 rw,relatime 0 0
```

Parte 2

Para este script hemos comprobado que nos pasan el parámetro, sino da error, una vez que lo tenemos ejecutamos este comando

```
ssh as@"$1" "sudo sfdisk -s && sudo sfdisk -l && sudo df -hT | grep -v -e 'udev' -e 'tmpfs'"
```

Se conecta por ssh a la máquina de la ip y ejecuta estos comandos:

`sudo sfdisk -s`: Muestra un resumen del tamaño total de todos los discos detectados en bloques.

`sudo sfdisk -l` para listar las particiones.

`sudo df -hT | grep -v -e 'udev' -e 'tmpfs'`: Este comando lista el uso del disco en formato legible (-h para "human-readable" y -T para mostrar el tipo de sistema de archivos), y filtra la salida para excluir las líneas que contienen 'udev' o 'tmpfs'.

Parte 3

3.1

Este script Bash está diseñado para extender un grupo de volúmenes existente en un sistema Linux que utiliza LVM (Logical Volume Manager).

Primero nos guardamos el nombre del grupo de volúmenes:

```
nombreVG="$1"
```

Luego con shift eliminamos el primer parámetro

Usamos `vgextend` para añadir el número de dispositivos que nos han pasado

```
sudo vgs $nombreVG
```

Tras extender el grupo de volúmenes, este comando muestra el estado actualizado del grupo de volúmenes especificado.

```
sudo pvs
```

Finalmente, muestra todos los dispositivos físicos para ver que se han añadido correctamente.

Parte 3.2

Este script de Bash está diseñado para manejar la creación y extensión de volúmenes lógicos en un sistema que utiliza el Logical Volume Manager (LVM).

```
rutaLV=$(lvsdisplay "$nombreGrupoVolumen/$nombreVolumenLogico" -Co "lv_path" | grep "$nombreGrupoVolumen/$nombreVolumenLogico")
```

Aquí, `lvsdisplay` es llamado con la opción `-Co` para mostrar sólo la columna `lv_path`, que proporciona la ruta completa al dispositivo del volumen lógico. El uso de `grep` filtra la salida

para asegurarse de que sólo se considera la línea correspondiente al volumen lógico específico.

El script entonces verifica si la variable rutaLV está vacía para determinar si el volumen lógico necesita ser creado o ya existe y, por tanto, necesita ser extendido. Si el volumen existe (if [-n "\$rutaLV"]), el script extiende el volumen lógico usando lvextend y luego redimensiona el sistema de archivos con resize2fs. Si el volumen no existe, se crea un nuevo volumen lógico con lvcreate. Luego, verifica si el directorio de montaje existe y, si no, lo crea con mkdir -p.

Una vez creado el volumen, se añade una entrada al archivo /etc/fstab para asegurar que el volumen se monte automáticamente en cada arranque del sistema.

Finalmente, el script formatea el nuevo volumen lógico con el sistema de archivos especificado y monta el volumen:

```
mkfs.$tipoSistemaFicheros $rutaLV  
mount $rutaLV $directorioMontaje
```