**1. Configuración del entorno**

**Antes de comenzar con el laboratorio, es necesario añadir un nuevo disco duro de 10 GB de tipo “balanceado” a la máquina virtual.**

sudo apt-get install fdisk

sudo apt-get install util-linux

Desde Google Cloud Console, crear el disco.

????????????????

systemctl daemon-reload

lsblk -e7

sudo blkid

sudo vim /etc/fstab

????????????????

**2. Gestión básica**

**En esta parte del laboratorio propone unas tareas de manipulación básica de particiones:**

**1) En el disco recién creado, crear 4 particiones de 1GB cada una y formatearlas: una de ellas será ext3, otra btrfs, otra xfs y última ext4.**

$ sudo cfdisk /dev/sdc

$ lsblk -e7

$ sudo mkfs.ext3 /dev/sdc1

$ sudo mkfs.btrfs /dev/sdc2

$ sudo mkfs.xfs /dev/sdc3

$ sudo mkfs.ext4 /dev/sdc4

**2) Configurar un montaje automático de las particiones (al arranque de la máquina virtual) en /disco[X] donde X es 1, 2, 3, 4. Se recomienda seguir estos pasos. Verificar que los montajes se mantienen al reiniciar la máquina. En caso de que la MV no arranque o no permita acceso vía SSH por problemas de configuración, se puede iniciar una terminal mínima llamada “consola serie” que provee acceso root para resolver incidencias.**

$ sudo mkdir -p /disco1 /disco2 /disco3 /disco4

$ sudo mount /dev/sdc1 /disco1

$ sudo mount /dev/sdc2 /disco2

$ sudo mount /dev/sdc3 /disco3

$ sudo mount /dev/sdc4 /disco4

$ sudo blkid

/dev/sdb15: SEC\_TYPE="msdos" UUID="8114-23C8" BLOCK\_SIZE="512" TYPE="vfat" PARTUUID="300ea48e-d3e0-d245-af8b-02496f983839"

/dev/sdb1: UUID="fd09b023-c217-4b53-a427-35b1910449d1" BLOCK\_SIZE="4096" TYPE="ext4" PARTUUID="5fe9d5fb-99ab-fb49-90fb-727dd13fbf5f"

/dev/sdc2: UUID="769fe467-7b66-4f19-b8a0-6c92202b089b" UUID\_SUB="1574eb5e-31e0-4079-b295-96afda2e4171" BLOCK\_SIZE="4096" TYPE="btrfs" PARTUUID="dc813d8e-bedc-0642-ba9c-c0a3b6433050"

/dev/sda2: UUID="689a4c4b-8df1-4db0-9bfa-15735acd88df" BLOCK\_SIZE="4096" TYPE="ext4" PARTUUID="bf26415e-8309-8249-8ba3-d6d5f0fb4267"

/dev/sda1: UUID="899a70b9-58b2-4364-be9e-231b3eb5d40a" BLOCK\_SIZE="4096" TYPE="ext4" PARTUUID="a35586f8-b0fe-034e-9aa8-c0abbcde4508"

/dev/sdc3: UUID="b9e24c43-579e-47d7-8041-4ff45620f585" BLOCK\_SIZE="4096" TYPE="xfs" PARTUUID="ff4ad6cf-94ac-d74b-b64e-29746e56e2ea"

/dev/sdc1: UUID="283a8256-f7fe-4a22-a24a-aeb595daaee2" BLOCK\_SIZE="4096" TYPE="ext3" PARTUUID="2ad25152-2448-bd4a-8a2c-11e799a395b2"

/dev/sdc4: UUID="7b52be38-60e4-4fe4-8ec7-376301b8cf28" BLOCK\_SIZE="4096" TYPE="ext4" PARTUUID="f67cf28e-af81-3546-9119-d0c91c90f2b9"

/dev/sdb14: PARTUUID="82e3a28a-7f26-0f4a-b499-84561ddd96f8"

$ sudo vim /etc/fstab

Hau gehitu:

UUID="283a8256-f7fe-4a22-a24a-aeb595daaee2" /disco1 ext3 defaults 0 0

UUID="769fe467-7b66-4f19-b8a0-6c92202b089b" /disco2 btrfs defaults 0 0

UUID="b9e24c43-579e-47d7-8041-4ff45620f585" /disco3 xfs defaults 0 0

UUID="7b52be38-60e4-4fe4-8ec7-376301b8cf28" /disco4 ext4 defaults 0 0

**3) Todos los sistemas de ficheros utilizan cierto espacio para almacenar meta-información. De entre los creados en el primer punto, ¿cuál de ellos utiliza más espacio?**

$ df -h /disco1 /disco2 /disco3 /disco4

Filesystem Size Used Avail Use% Mounted on

/dev/sdc1 975M 60K 924M 1% /disco1

/dev/sdc2 1.0G 5.8M 905M 1% /disco2

/dev/sdc3 960M 40M 921M 5% /disco3

/dev/sdc4 974M 24K 907M 1% /disco4

Usualmente, btrfs utiliza más espacio para metadatos que ext3/ext4/xfs.

**4) ¿Es posible acceder a una partición ext3 que ha sido montada como ext4? ¿Y al revés? ¿Por qué?**

Se puede montar una partición ext3 como ext4 (ext4 es compatible hacia atrás), pero montar una ext4 como ext3 puede causar errores si se usaron funciones ext4

**5) Desmontar y borrar las 3 últimas particiones. Crear una única partición ext4 de 8 GB.**

$ sudo umount /disco1 /disco2 /disco3 /disco4

$ sudo cfdisk /dev/sdc

\*\* Borratu ta sortu \*\*

$ sudo mkfs.ext4 /dev/sdc2

$ sudo mount /dev/sdc2 /disco2 → ?

**6) Copiar el contenido del directorio /var en la nueva partición ext4 que acabas de crear. Después, redimensionar la partición para que sea lo más pequeña posible. Recordar que, al modificar una partición, no tiene por qué modificarse el sistema de ficheros que contiene.**

$ sudo rsync -avh /var/ /disco2/

$ sudo umount /disco2

$ sudo e2fsck -f /dev/sdc2

$ sudo resize2fs -M /dev/sdc2

**7) Eliminar la configuración de montaje automático realizada en el 2º paso.**

$ sudo vim /etc/fstab

* Eliminar las líneas.

**3. Comparativa de rendimiento**

**Google Cloud ofrece diferentes tipos de discos que se pueden usar en las máquinas virtuales de Compute Engine. La documentación indica el precio asociado a cada tipo de disco, pero no se detallan valores concretos de rendimiento.**

**Para poder conocer las capacidades de los diferentes discos, se pueden utilizar herramientas de benchmark que realizan pruebas de stress para caracterizar el rendimiento. Estas pruebas son útiles para conocer si un disco (o sistema de ficheros) está siendo un cuello de botella para las aplicaciones en uso. En este apartado se proponen diferentes tareas con el benchmark fio:**

**1) Borrar las particiones creadas en el disco en la sección anterior. En este ejercicio nos referiremos a este disco como balanceado.**

$ sudo cfdisk /dev/sdb

**2) Añadir un nuevo disco de 10 GB de tipo “SSD” a la máquina virtual. En este ejercicio nos referiremos a este disco como SSD.**

$ sudo cfdisk /dev/sdc

**3) En ambos discos, crear 1 partición de 4 GB y formatearla como Ext4. Montar la partición del disco balanceado en el directorio /discoBalanceado y la partición del disco SSD en el directorio /discoSSD. No es necesario configurar los montajes para que se realicen en el arranque del sistema.**

$ sudo mkfs.ext4 /dev/sdb1

$ sudo mkfs.ext4 /dev/sdc1

$ sudo mkdir /discoBalanceado /discoSSD

$ sudo mount /dev/sdb1 /discoBalanceado

$ sudo mount /dev/sdc1 /discoSSD

**4) Revisar la 1ª sección de este artículo3 para aprender a instalar y realizar una prueba de rendimiento con fio.**

**5) Instalar fio en el sistema siguiendo el comando que se indica en el artículo.**

$ sudo apt update

$ sudo apt install fio

**6) Utilizar fio para realizar una medición de las IOPS de escritura mediante la realización de operaciones de escritura aleatorias, con las siguientes características:**

**a. Tamaño de bloque de E/S: 4 KB**

**b. Profundidad de E/S: Al menos 256**

**c. Tamaño del fichero de contenido aleatorio de 1 GB.**

**d. Duración de la prueba: 1 minuto.**

**e. Resto de parámetro por defecto (cómo indicados en el ejemplo del artículo).**

**Esta prueba se debe realizar con ambos discos. Los valores más representativos de las pruebas son las métricas de velocidad de lectura (READ) y escritura (WRITE), que se encuentran entre las últimas líneas que genera fio. Estos valores se indican en kB/s o MB/s, ¿qué diferencia hay entre ambos discos?**

$ cd /discoBalanceado

$ fio --name=write\_test --ioengine=libaio --rw=randwrite --bs=4k --size=1G --numjobs=1 --iodepth=256 --runtime=60 --time\_based

cd /discoSSD

$ fio --name=write\_test --ioengine=libaio --rw=randwrite --bs=4k --size=1G --numjobs=1 --iodepth=256 --runtime=60 --time\_based

* Anota los valores **WRITE (IOPS, BW)** al final.

**7) Repetir la prueba anterior, esta vez para medir las IOPS de de lectura mediante la realización de operaciones de lecturas aleatorias. Utilizar los mismos parámetros que para la prueba anterior. ¿Varían los resultados respecto a la 1ª prueba?**

$ cd /discoBalanceado

$ fio --name=write\_test --ioengine=libaio --rw=randread --bs=4k --size=1G --numjobs=1 --iodepth=256 --runtime=60 --time\_based

cd /discoSSD

$ fio --name=write\_test --ioengine=libaio --rw=randread --bs=4k --size=1G --numjobs=1 --iodepth=256 --runtime=60 --time\_based

* Berdina, baina --rw=randread erabilita.

**8) Revisar el precio de los discos de tipo balanceado y SSD en Google Cloud, para la región que estéis usando. ¿La diferencia de precio es proporcional con la diferencia de rendimiento?**

El SSD debería tener mayores IOPS y ancho de banda (MB/s) que el balanceado.

Precio ≠ rendimiento lineal. El SSD es más caro, pero el rendimiento suele ser mucho mayor.

**9) Desmontar y eliminar el disco de tipo SSD.**

**4. Gestión básica**

**En esta parte del laboratorio se trabaja con LVM y RAID:**

**1) Borrar las particiones creadas en las secciones anteriores en el disco.**

$ sudo cfdisk /dev/sdb

$ sudo cfdisk /dev/sdc

**2) Añadir un nuevo disco de 10 GB de tipo balanceado a la máquina virtual.**

$ sudo cfdisk /dev/sdb

$ sudo cfdisk /dev/sdd

**3) Crear 2 particiones de 3 GB en cada disco. Crear un volumen lógico LVM con 3 de las 4 particiones. Crear una partición ext4 en el volumen usando el 100% de espacio.**

$ sudo apt install lvm2

$ sudo pvcreate /dev/sdb1 /dev/sdb2 /dev/sdd1

$ sudo vgcreate vgdatos /dev/sdb1 /dev/sdb2 /dev/sdd1

$ sudo lvcreate -l 100%FREE -n lvdatos vgdatos

$ sudo mkfs.ext4 /dev/vgdatos/lvdatos

$ sudo mkdir /mnt/lvm

$ sudo mount /dev/vgdatos/lvdatos /mnt/lvm

**4) Montar el sistema de ficheros en un directorio y comprobar su estado. Para ello, copiar alguno de los directorios del sistema operativo a él.**

$ sudo cp -r /etc /mnt/lvm/

$ df -h /mnt/lvm

**5) Añadir la 4ª partición al volumen lógico y extender el tamaño del sistema de ficheros para que ocupe el total del volumen. Comprobar que los datos copiados en el paso anterior siguen estando.**

$ sudo vgextend vgdatos /dev/sdd2

$ sudo lvextend -l +100%FREE /dev/vgdatos/lvdatos

$ sudo resize2fs /dev/vgdatos/lvdatos

$ ls /mnt/lvm/etc

**6) Borrar el volumen lógico recién creado (utilizar los comandos LVM apropiados).**

$ sudo umount /mnt/lvm

$ sudo lvremove /dev/vgdatos/lvdatos

$ sudo vgremove vgdatos

$ sudo pvremove /dev/sd{b1,b2,d1,d2}

**7) Crear un sistema RAID 5 con 3 de las particiones. Crear un sistema de ficheros ext4 para el sistema RAID 5 y hacerlo accesible. Copiar el contenido de la carpeta /var a la carpeta del sistema RAID.**

$ sudo apt install mdadm -y

$ sudo mdadm --create /dev/md0 --level=5 --raid-devices=3 /dev/sdb1 /dev/sdb2 /dev/sdd1

$ sudo mkfs.ext4 /dev/md0

$ sudo mkdir /mnt/raid5

$ sudo mount /dev/md0 /mnt/raid5

$ sudo rsync -avh /var/ /mnt/raid5/

**8) Simular un fallo en el tercer disco (parámetro -f). Recuperar la información perdida usando la partición que quedó libre.**

$ sudo mdadm /dev/md0 -f /dev/sdd1

$ cat /proc/mdstat

$ sudo mdadm /dev/md0 -r /dev/sdd1

$ sudo mdadm /dev/md0 -a /dev/sdd2

$ cat /proc/mdstat

**9) Desmontar y eliminar el dispositivo RAID. Eliminar el disco creado en el paso 2.**

$ sudo umount /mnt/raid5

$ sudo mdadm --stop /dev/md0

$ sudo mdadm --zero-superblock /dev/sd{b1,b2,d1,d2}

**5. Copias de seguridad**

**En esta última parte del laboratorio se trabaja con la herramienta rsnapshot:**

**1) Borrar las particiones creadas en la sección anterior en el disco.**

**2) Crear 1 partición de 4 GB en el disco y formatearla como ext4. Montarla en un directorio llamado /backups.**

$ sudo cfdisk /dev/sdb

$ sudo mkfs.ext4 /dev/sdb1

$ sudo mkdir /backups

$ sudo mount /dev/sdb1 /backups

**3) Instalar rsnapshot en el sistema y revisar este4 documento donde se detalla su configuración.**

$ sudo apt install rsnapshot

$ sudo cp /etc/rsnapshot.conf /etc/rsnapshot.conf.bak

$ sudo vim /etc/rsnapshot.conf

HOR, HAU IDATZI:  
snapshot\_root /backups/

retain hourly 24

retain daily 7

retain weekly 4

backup /home/ localhost/

backup /etc/ localhost/

backup /var/log/ localhost/

& sudo rsnapshot configtest

**4) Configurar rsnapshot de la siguiente forma:**

**a. Directorio para almacenar las copias de seguridad: /backups.**

**b. Niveles de copia e intervalos:**

**i. “horaria”, 24**

**ii. “diaria”, 7**

**iii. “semanal”, 4**

**c. Directorios a guardar (todos se almacenan en el directorio /backups): /home, /etc y /var/log**

$ sudo cp /etc/rsnapshot.conf /etc/rsnapshot.conf.bak

$ sudo vim /etc/rsnapshot.conf

HOR, HAU IDATZI:  
snapshot\_root /backups/

retain hourly 24

retain daily 7

retain weekly 4

backup /home/ localhost/

backup /etc/ localhost/

backup /var/log/ localhost/

**5) Verificar que la configuración es correcta con el comando rsnapshot configtest.**

$ sudo rsnapshot configtest

**6) Realizar una copia de tipo “horaria” y revisar que los contenidos se han copiado correctamente.**

$ sudo rsnapshot hourly

$ ls /backups/hourly.0

**7) Crear una carpeta y un fichero nuevo en el directorio /home de tu usuario (incluye algo de texto en el fichero). Después, realizar una nueva copia de tipo “horaria”.**

$ echo “Proba testua” > ~/nuevo.txt

$ sudo rsnapshot hourly

**8) Verificar que la nueva copia se ha hecho correctamente y revisar los cambios entre ambas copias con el comando rsnapshot-diff.**

$ sudo rsnapshot-diff /backups/hourly.1/ /backups/hourly.0/