

Bootcamp Devops Engineer

Encuentro #3

Antes de empezar... quiz!

Que vamos a ver hoy?

- Sistema de Archivos
- Particionado de Discos o Volúmenes
- Mantenimiento de Sistema de Archivos



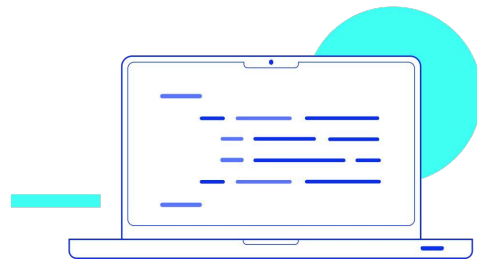
Sistema de Archivos



¿Qué es un sistema de archivos?

Un conjunto de páginas apiladas no es un libro. Los índices y los números de páginas lo convierten en un libro. Con un sistema de archivos pasa algo parecido. **Un sistema de archivos sirve para poder localizar la información de manera coherente.** A diferencia de un libro necesitamos esa coherencia, no solamente para “leer” sino para “escribir”, es decir para modificar la información almacenada.

Ext2, ext3 y ext4 son sistemas de archivos creados para **Linux**. En los próximos slides, veremos las diferencias entre estos sistemas de ficheros.



El más usado en AWS: ext4

AWS ofrece una variedad de sistemas de archivos para sus instancias de EC2 y volúmenes de EBS, pero ext4 es el predeterminado y más comúnmente usado debido a su balance entre rendimiento, estabilidad y características.

1. **Estabilidad:** ext4 es una versión mejorada de ext3, que a su vez es una versión mejorada de ext2. Ha sido utilizado y probado extensamente en producción durante muchos años, lo que lo hace muy estable y confiable.
2. **Rendimiento:** ext4 ofrece un buen rendimiento en la mayoría de los casos de uso, especialmente en operaciones de lectura/escritura y en la gestión de grandes cantidades de archivos pequeños.
3. **Journaling:** Utiliza un sistema de journaling que ayuda a proteger la integridad de los datos en caso de fallos inesperados del sistema, mejorando la durabilidad y la seguridad de los datos.
4. **Amplia Compatibilidad:** ext4 es ampliamente soportado en prácticamente todas las distribuciones de Linux, lo que lo hace una opción segura para entornos heterogéneos.
5. **Soporte para Grandes Volúmenes:** ext4 soporta volúmenes de hasta 1 exabyte y archivos individuales de hasta 16 terabytes, lo cual es más que suficiente para la mayoría de las aplicaciones.

Otros sistemas de archivos

1. **XFS:** Conocido por su alto rendimiento en operaciones de entrada/salida intensivas y su capacidad para manejar grandes volúmenes de datos. Es una buena opción para aplicaciones que requieren un alto rendimiento de escritura, como bases de datos y sistemas de big data.
2. **Btrfs (B-tree File System):** Aunque menos común, Btrfs se utiliza para sus capacidades avanzadas, como instantáneas y verificación de integridad. Es adecuado para casos de uso donde la integridad de los datos y la gestión de versiones son críticos.
3. **EFS (Elastic File System):** Un sistema de archivos administrado por AWS que se escala automáticamente y permite el acceso concurrente desde múltiples instancias EC2. Es ideal para aplicaciones que necesitan almacenamiento compartido y escalable.
4. **EBS (Elastic Block Store):** Aunque no es un sistema de archivos per se, los volúmenes de EBS se utilizan para almacenar datos y pueden ser formateados con cualquier sistema de archivos soportado por Linux, como ext4, XFS o Btrfs.

Práctica

Iniciar nuestra instancia Linux

- Crear la instancia - `multipass launch -n <nombre>`
- Tomar la shell de nuestra instancia - `multipass shell <nombre>`

Clonar repositorio de la clase

- `git clone https://github.com/JCaimo/Clase3.git`

Pasos para crear, particionar y montar un volumen

- Crea un archivo de imagen de disco: `sudo dd if=/dev/zero of=/mnt/data.img bs=1M count=1024`
- Asocia el archivo a un dispositivo de bloque: `sudo losetup -fP /mnt/data.img`
- Verifica el dispositivo de bloque: `losetup -a AND lsblk`
- Crea una partición: `sudo fdisk /dev/loop0`
- Formatear la partición: `sudo mkfs.ext4 /dev/loop0p1`
- Montar la partición: `sudo mkdir /mnt/data AND sudo mount /dev/loop0p1 /mnt/data`
- Verificar el montaje: `df -h AND lsblk`
- Cómo automatizamos este proceso?

Mantenimiento de Sistemas de Archivos



Espacio en los inodos libres de particiones

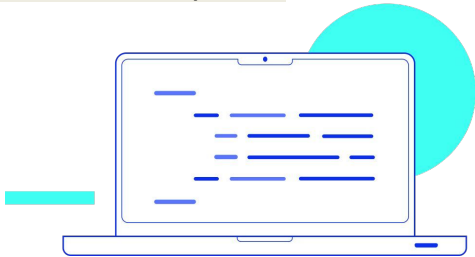
Un sistema de lectura/escritura no sirve de mucho si crece hasta el punto en que no pueda admitir nuevos ficheros. Esto puede ocurrir si nuestro sistema de ficheros se llena o si se queda sin inodos libres. Cada sistema de archivos contiene un número finito de inodos que se establecen en el momento de creación del sistema de archivos.

Este número es, a su vez, el máximo número de archivos que un sistema de archivos puede acomodar. Es posible quedarse sin inodos libres en particiones que contengan muchos ficheros pequeños.



Comandos file, df, du y sus argumentos

- Ver el tipo de sistema de archivos: `file -sL /dev/sda1`
- Mostrar inodos libres: `df -i`
- Mostrar espacio: `df -h`
- Mostrar listado de particiones, espacio ocupado y punto de montaje: `df -h | grep '^/dev' | tr -s ' ' | cut -d' ' -f1,2,6`
- Espacio total ocupado por /etc: `du -sch /etc`
- Espacio total ocupado por distintos directorios: `du -sch /*`
- Traer los 10 archivos más grandes en el directorio actual: `du -sm * | sort -nr | head`
- Verificar el espacio ocupado por un /home y sus subdirectorios: `du -h --max-depth=1 /home`



Recomendaciones

- Podcasts:
 - Inglés: The DevOps Paradox
 - Español: Enchiladas DevOps
- Youtube:
 - <https://www.youtube.com/@DevOpsToolkit>

