

Aquí tienes un documento completo sobre BTRFS, sus características, comparativas y un ejercicio práctico:

BTRFS: Un Sistema de Ficheros de Nueva Generación para Linux

BTRFS (abreviatura de B-tree File System) es un sistema de ficheros de copia en escritura (CoW o Copy-on-Write) para Linux, que fue diseñado con el objetivo de implementar características avanzadas y robustas, combinando algunas de las mejores ideas de ZFS con un enfoque de código abierto. Su desarrollo comenzó en Oracle en 2007 y ha ido ganando tracción como una alternativa moderna a los sistemas de ficheros tradicionales como ext4 o XFS.

Características Principales de BTRFS

BTRFS se distingue por una serie de características innovadoras que lo hacen atractivo para diversos casos de uso:

- **Copy-on-Write (CoW):** Esta es la característica fundamental de BTRFS. Cuando se realiza una modificación en un bloque de datos, BTRFS no sobrescribe el bloque original, sino que escribe los datos modificados en un nuevo bloque. Esto es crucial para la integridad de los datos, ya que evita la corrupción de los datos en caso de un fallo del sistema. Además, facilita la creación de instantáneas (snapshots).
- **Instantáneas (Snapshots):** Las instantáneas son una de las características más potentes de BTRFS. Permiten crear "fotos" del estado del sistema de ficheros en un momento dado, casi instantáneamente y sin consumir espacio adicional significativo. Dado que son CoW, solo se almacenan los bloques modificados desde la instantánea. Esto es invaluable para copias de seguridad, rollbacks de sistema o pruebas de software.
- **Subvolúmenes:** BTRFS no utiliza particiones tradicionales para organizar los datos. En su lugar, utiliza subvolúmenes, que son puntos de montaje flexibles dentro de un sistema de ficheros BTRFS. Cada subvolumen tiene su propio espacio de nombres y puede ser montado de forma independiente. Esto permite una gran flexibilidad en la organización del disco y en la gestión de instantáneas.
- **Integridad de Datos (Checksums):** BTRFS calcula sumas de verificación (checksums) para los datos y los metadatos, garantizando la integridad de la información almacenada. Si se detecta una corrupción, BTRFS puede intentar reparar el problema utilizando copias redundantes de los datos (si están configuradas).
- **Gestión de Volumen Integrada (RAID):** BTRFS puede gestionar múltiples dispositivos de almacenamiento (discos) en un único sistema de ficheros, ofreciendo funcionalidades de RAID (RAID0, RAID1, RAID10, RAID5, RAID6). Esto permite combinar la capacidad de varios discos y, en algunos modos RAID, proporciona redundancia para proteger contra fallos de disco.
- **Compresión Transparente:** BTRFS puede comprimir los datos de forma transparente antes de escribirlos en el disco, lo que puede ahorrar espacio y mejorar el rendimiento en ciertas cargas de trabajo. Soporta algoritmos como zlib, LZO y ZSTD.

- **Deduplicación (Experimental/Offline):** Aunque no es una característica central y suele requerir herramientas externas, BTRFS tiene la capacidad de deduplicar bloques de datos idénticos para ahorrar espacio.
- **Redimensionamiento en Línea:** Los sistemas de ficheros BTRFS y los subvolumenes pueden ser redimensionados dinámicamente, incluso mientras están en uso, sin necesidad de desmontarlos.

BTRFS vs. Otros Sistemas de Ficheros

A continuación, se presenta una comparación de BTRFS con algunos de los sistemas de ficheros más comunes:

Característica	BTRFS	Ext4	XFS	ZFS
Tipo	Copy-on-Write (CoW)	Journaling Filesystem	Journaling Filesystem	Copy-on-Write (CoW)
Instantáneas	Nativas (CoW, eficientes)	No nativas (requiere LVM o herramientas ext.)	No nativas (requiere LVM o herramientas ext.)	Nativas (CoW, muy robustas)
Checksums	Sí (datos y metadatos)	No	No	Sí (datos y metadatos, muy robustas)
RAID integrado	Sí (0, 1, 10, 5, 6 - con limitaciones en 5/6)	No (requiere mdadm)	No (requiere mdadm)	Sí (ZFS RAID-Z, muy robustas)
Compresión	Transparente (zlib, LZO, ZSTD)	No nativa	No nativa	Transparente (gzip, lz4, zstd)
Subvolumenes	Sí	No (directorios normales)	No (directorios normales)	Sí (Datasets)
Redimensionamiento	En línea (añadir/quitar dispositivos)	En línea (ampliar) / Offline (reducir)	En línea (ampliar) / Offline (reducir)	En línea (añadir/quitar dispositivos)

Característica	BTRFS	Ext4	XFS	ZFS
Deduplicación	Experimental / Offline	No	No	En línea (opcional, recursos intensivos)
Recuperación	Robusta (CoW, checksums)	Generalmente buena	Generalmente buena	Muy robusta (CoW, checksums, múltiples copias)
Madurez	Estable para la mayoría de usos (excepto RAID5/6)	Muy maduro y estable	Muy maduro y estable	Muy maduro y estable (Solaris/FreeBS D origen)

Pros de BTRFS:

- Protección de datos: CoW y checksums ofrecen una excelente protección contra la corrupción de datos.
- Flexibilidad: Instantáneas, subvolumenes y gestión de volumen integrada proporcionan una gran flexibilidad para la administración del almacenamiento.
- Eficiencia de espacio: Compresión y instantáneas eficientes pueden ahorrar una cantidad significativa de espacio en disco.
- Facilidad de administración: Muchas tareas de administración se simplifican gracias a las características integradas.
- Desarrollo activo: La comunidad de BTRFS está muy activa, y el sistema de ficheros sigue evolucionando.

Contras de BTRFS:

- Complejidad: Puede ser más complejo de entender y administrar inicialmente para usuarios no familiarizados con sus conceptos.
- Rendimiento en ciertas cargas: En algunas cargas de trabajo específicas (ej. muchas escrituras aleatorias pequeñas), el rendimiento puede ser inferior a ext4 o XFS debido a la naturaleza CoW.
- RAID5/6: Aunque existen, los modos RAID5 y RAID6 en BTRFS han tenido problemas históricos de estabilidad y rendimiento, y aún no se consideran tan robustos como las implementaciones de ZFS o mdadm. Se recomienda precaución al usarlos en producción.
- Recuperación de errores: Aunque la protección de datos es alta, la recuperación en caso de errores graves puede ser un desafío si no se tienen los conocimientos adecuados.

Subcomandos de btrfs

La herramienta principal para interactuar con BTRFS es el comando `btrfs`, que tiene una gran cantidad de subcomandos para gestionar volúmenes, subvolúmenes, instantáneas, etc.

Subcomando	Descripción	Ejemplo
<code>btrfs filesystem</code>	Subcomandos para gestionar sistemas de ficheros BTRFS (espacio, dispositivos, etc.).	<code>btrfs filesystem show</code> (mostrar sistemas de ficheros BTRFS)
<code>btrfs device</code>	Subcomandos para gestionar los dispositivos físicos que componen un sistema de ficheros BTRFS.	<code>btrfs device add /dev/sdb /mnt/btrfs</code> (añadir un disco a un RAID)
<code>btrfs subvolume</code>	Subcomandos para crear, listar, eliminar y gestionar subvolúmenes e instantáneas.	<code>btrfs subvolume create /mnt/btrfs/home</code> (crear un subvolumen)
<code>btrfs send/receive</code>	Para enviar y recibir instantáneas (útil para copias de seguridad incrementales).	<code>`btrfs send /mnt/btrfs/@snapshot_old</code>
<code>btrfs balance</code>	Redistribuye los datos y metadatos entre los dispositivos y cambia la configuración de RAID.	<code>btrfs balance start /mnt/btrfs</code> (equilibrar el sistema de ficheros)
<code>btrfs scrub</code>	Verifica la integridad de los datos en el sistema de ficheros (similar a <code>fsck</code> para otros sistemas).	<code>btrfs scrub start /mnt/btrfs</code> (iniciar una verificación)
<code>btrfs property</code>	Gestiona las propiedades de los objetos de BTRFS (subvolúmenes, instantáneas).	<code>btrfs property get /mnt/btrfs/data ro</code>
<code>btrfs quota</code>	Gestiona las cuotas de espacio en los subvolúmenes.	<code>btrfs quota enable /mnt/btrfs</code>
<code>btrfs inspect-internal</code>	Herramientas internas para inspeccionar la estructura de	<code>btrfs inspect-internal dump-tree /dev/sdb</code>

Subcomando	Descripción	Ejemplo
	BTRFS (para expertos).	
<code>btrfs replace</code>	Reemplaza un dispositivo defectuoso en un sistema de ficheros BTRFS.	<code>btrfs replace start 1 /dev/sdb /mnt/btrfs</code> (reemplazar disco ID 1)

Prueba Práctica: Creación y Gestión de BTRFS con Subvolúmenes

Objetivo: El objetivo de esta práctica es familiarizarse con la creación de un sistema de ficheros BTRFS multi-dispositivo (RAID1), montarlo, y crear y gestionar subvolúmenes para diferentes propósitos.

Discos a utilizar: /dev/sdb, /dev/sdc, /dev/sdd

Pre-requisitos:

- Acceso como `root` o con `sudo`.
- Los dispositivos /dev/sdb, /dev/sdc, /dev/sdd deben estar vacíos y no contener particiones o sistemas de ficheros importantes. ¡ADVERTENCIA: ESTA OPERACIÓN BORRARÁ CUALQUIER DATO EN ESOS DISCOS!
- Tener instalado el paquete `btrfs-progs`. En sistemas basados en Debian/Ubuntu: `sudo apt install btrfs-progs`. En Fedora/RHEL: `sudo dnf install btrfs-progs`.

Pasos:

1. Identificar y Limpiar los Discos (Paso Crucial)

Primero, asegúrate de que los discos son los correctos y limpia cualquier dato o partición existente.

```
# Listar los discos para confirmar que son los correctos
sudo fdisk -l /dev/sdb /dev/sdc /dev/sdd
```

```
# Limpiar las tablas de particiones existentes (MUY IMPORTANTE Y DESTRUCTIVO)
# Repetir para sdb, sdc, sdd
sudo wipefs -a /dev/sdb
sudo wipefs -a /dev/sdc
sudo wipefs -a /dev/sdd
```

Observación: `wipefs -a` eliminará todas las firmas de sistemas de ficheros y tablas de particiones. Ten extrema precaución.

2. Crear el Sistema de Ficheros BTRFS (RAID1)

Crearemos un sistema de ficheros BTRFS utilizando /dev/sdb y /dev/sdc en modo RAID1 para datos y metadatos. Dejaremos /dev/sdd para añadirlo más tarde o para otros propósitos.

```
# Crear el sistema de ficheros BTRFS en RAID1
# -d raid1: modo RAID para los datos (RAID1)
# -m raid1: modo RAID para los metadatos (RAID1)
sudo mkfs.btrfs -d raid1 -m raid1 /dev/sdb /dev/sdc
```

Salida esperada:

```
btrfs-progs vX.Y.Z
See http://btrfs.wiki.kernel.org for more information.
```

```
Performing full device TRIM (to remove any stale filesystem data)
... (puede tardar un poco) ...
Detected a free space tree in the device superblock, cleaning it up now
Detected a free space tree in the device superblock, cleaning it up now
uuid: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
Label: (none)
Nodesize: 16384
...
```

Apunta el `uuid` que te muestra la salida, lo usaremos para montar el sistema de ficheros.

3. Montar el Sistema de Ficheros BTRFS

Creamos un punto de montaje y montamos el sistema de ficheros.

```
# Crear el punto de montaje
sudo mkdir /mnt/btrfs_data

# Montar el sistema de ficheros BTRFS usando el UUID (más robusto que el nombre
del dispositivo)
# Reemplaza 'XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX' con el UUID real de tu
mkfs.btrfs
sudo mount -o defaults,compress=zstd,ssd /dev/sdb /mnt/btrfs_data
```

Explicación de las opciones de montaje:

- `defaults`: Opciones de montaje por defecto.
- `compress=zstd`: Habilita la compresión ZSTD. Es una buena opción que equilibra compresión y rendimiento.
- `ssd`: Optimiza BTRFS para unidades de estado sólido (SSD) al ajustar el algoritmo de asignación. Si usas HDDs, omite esta opción.

4. Verificar el Sistema de Ficheros Montado

```
# Verificar que el sistema de ficheros BTRFS está montado y en RAID1
sudo btrfs filesystem show /mnt/btrfs_data

# Verificar el uso del espacio
sudo btrfs filesystem usage /mnt/btrfs_data
```

Salida esperada de `btrfs filesystem show` (ejemplo):

```
Label: none
uuid: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
    Total devices 2 FS bytes used 16.00MiB
    devid    1 size 10.00GiB used 2.00GiB path /dev/sdb
    devid    2 size 10.00GiB used 2.00GiB path /dev/sdc
```

Salida esperada de `btrfs filesystem usage` (ejemplo):

```
Overall:
    Device size:          20.00GiB
    Device allocated:      4.00GiB
    Device unallocated:    16.00GiB
    ...
```

```
Data, RAID1:      2.00GiB
Metadata, RAID1:  1.00GiB
System, RAID1:    8.00MiB
...
```

5. Crear Subvolúmenes

Crearemos algunos subvolúmenes para organizar nuestros datos. Por ejemplo: `home_users`, `web_data`, `backups`.

```
# Crear el subvolumen para usuarios
sudo btrfs subvolume create /mnt/btrfs_data/home_users

# Crear el subvolumen para datos web
sudo btrfs subvolume create /mnt/btrfs_data/web_data

# Crear el subvolumen para copias de seguridad
sudo btrfs subvolume create /mnt/btrfs_data/backups
```

6. Listar Subvolúmenes

```
# Listar todos los subvolúmenes en el sistema de ficheros
sudo btrfs subvolume list /mnt/btrfs_data
```

Salida esperada (ejemplo):

```
ID 256 gen 48 top level 5 path home_users
ID 257 gen 49 top level 5 path web_data
ID 258 gen 50 top level 5 path backups
```

7. Montar Subvolúmenes Específicos (Opcional)

Puedes montar subvolúmenes individualmente en diferentes puntos de montaje si lo deseas. Esto es útil para separar el árbol de directorios raíz de un subvolumen específico.

```
# Crear puntos de montaje para los subvolúmenes
sudo mkdir /mnt/home /mnt/web

# Montar el subvolumen home_users
# Usa el UUID del sistema de ficheros BTRFS, no del subvolumen
sudo mount -o defaults,compress=zstd,ssd,subvol=home_users /dev/sdb /mnt/home

# Montar el subvolumen web_data
sudo mount -o defaults,compress=zstd,ssd,subvol=web_data /dev/sdb /mnt/web
```

Ahora, si creas archivos en `/mnt/home`, estarán en el subvolumen `home_users`.

8. Crear una Instantánea (Snapshot)

Crearemos una instantánea del subvolumen `home_users`. Esto es muy útil antes de realizar cambios importantes.

```
# Crear algunos archivos de prueba en home_users
```



```
sudo touch /mnt/home/file1.txt
sudo echo "Hello from home" | sudo tee /mnt/home/test.log

# Crear una instantánea de home_users
# La instantánea es de solo lectura por defecto. Para que sea de escritura,
añade -r.
sudo btrfs subvolume snapshot /mnt/btrfs_data/home_users
/mnt/btrfs_data/snapshots/home_users_20250704_initial
```

9. Explorar la Instantánea

```
# Verificar que los archivos están en la instantánea
sudo ls /mnt/btrfs_data/snapshots/home_users_20250704_initial
```

Ahora, si modificas o eliminas archivos en `/mnt/home`, la instantánea `home_users_20250704_initial` seguirá conteniendo el estado original.

10. Añadir un Nuevo Dispositivo al Sistema de Ficheros (Escalar RAID1)

Ahora añadiremos `/dev/sdd` a nuestro sistema de ficheros BTRFS. Esto no cambiará el modo RAID actual (RAID1 en `sdb/sdc`), pero permitirá a BTRFS usar el espacio adicional si decidimos reequilibrar o cambiar la configuración de RAID.

```
# Añadir el nuevo dispositivo al sistema de ficheros BTRFS
sudo btrfs device add /dev/sdd /mnt/btrfs_data

# Verificar los dispositivos en el sistema de ficheros
sudo btrfs filesystem show /mnt/btrfs_data
```

Salida esperada (ejemplo):

```
Label: none
uuid: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
    Total devices 3 FS bytes used 16.00MiB
    devid    1 size 10.00GiB used 2.00GiB path /dev/sdb
    devid    2 size 10.00GiB used 2.00GiB path /dev/sdc
    devid    3 size 10.00GiB used 0.00B path /dev/sdd # sdd con 0.00B usado
aún
```

Observa que `/dev/sdd` se ha añadido, pero aún no tiene datos.

11. Reequilibrar el Sistema de Ficheros (Balance)

Para que BTRFS comience a usar el espacio del nuevo disco y redistribuya los datos según el modo RAID configurado, necesitamos realizar un **balance**. En este caso, como los datos están en RAID1, BTRFS puede copiar bloques existentes a `/dev/sdd` para mantener la redundancia en tres discos si se cambia el perfil de datos o para mejorar la distribución de bloques.

```
# Reequilibrar el sistema de ficheros. Esto puede llevar tiempo.
sudo btrfs balance start /mnt/btrfs_data
```

Puedes ver el progreso del balance en otra terminal con:

```
sudo btrfs balance status /mnt/btrfs_data
```

12. Verificar Después del Balance

```
sudo btrfs filesystem usage /mnt/btrfs_data
sudo btrfs filesystem show /mnt/btrfs_data
```

Ahora deberías ver que `/dev/sdd` ha empezado a ser utilizado para almacenar datos (aunque en RAID1 solo usará espacio equivalente al más grande de los otros discos si es un nuevo espejo).

13. Limpieza (Opcional, para deshacer la práctica)

Si quieres limpiar los puntos de montaje y los discos para futuras prácticas:

```
# Desmontar subvolúmenes
sudo umount /mnt/home
sudo umount /mnt/web

# Desmontar el sistema de ficheros principal
sudo umount /mnt/btrfs_data

# Eliminar los subvolúmenes y la instantánea
# Deben estar desmontados
sudo btrfs subvolume delete /mnt/btrfs_data/home_users
sudo btrfs subvolume delete /mnt/btrfs_data/web_data
sudo btrfs subvolume delete /mnt/btrfs_data/backups
sudo btrfs subvolume delete
/mnt/btrfs_data/snapshots/home_users_20250704_initial

# Eliminar el punto de montaje principal
sudo rm -r /mnt/btrfs_data
sudo rm -r /mnt/home /mnt/web

# Limpiar las firmas de BTRFS de los discos
# ¡MUCHO CUIDADO! Esto borrará el BTRFS creado
sudo wipefs -a /dev/sdb
sudo wipefs -a /dev/sdc
sudo wipefs -a /dev/sdd
```

Con esta práctica, habrás creado un sistema de ficheros BTRFS multi-dispositivo, gestionado subvolúmenes y creado una instantánea, familiarizándote con algunas de las características clave de BTRFS.