Learning Objectives

Learners will be able to...

deploy the following tools for Filesystem Management:
- XFS tools
- Ext4 tools
- Btrfs tools

Filesystems

A file system controls how data is written to, and read from, the location where it is stored. We will explore three popular Linux systems. The user experience on each of them is similar but their underlying features may differ.

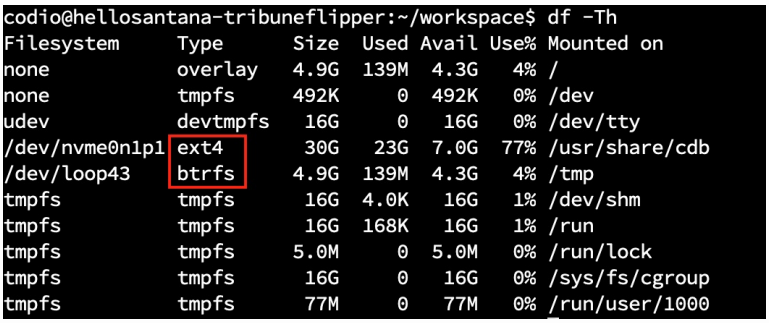
A feature often discussed when talking about file systems is **journaling**. The term refers to a log that is kept during disk writing that can be used to reconstruct data that might get corrupted due to a power outage or system crash.

What filesystem are we running?

You can run the `df` command below to get that information. The command `df` is short for "disk filesystem". The options we are passing the command are `-T` for type and `h` displays the output in a more human-readable form.

```
df -Th
```

Your output should look something like the image below.



Output of the df -Th command showing file systems being used. There is a red box around ext4 and btrfs

Most popular Linux file systems

Ext4

Ext4 is the most popular Linux file system. It is the 4th version of the extended filesystem and known to be very stable. It is backward compatible with previous versions. It is a journaled system.

Tools for managing the **Ext4** file system

Tool Name	Purpose
`dumpe2fs`	Display filesystem information
`e2fsck`	Examine the filesystem for errors
`fsck`	Fix filesystem errors
`tune2fs`	Change tunable parameters

XFS

The **XFS** file system is known for performance and its ability to handle large files. XFS uses metadata journaling as opposed to journaling both the metadata and data as Ext4 does. This leads to faster performance but can result in loss of data in cases of power loss.

Tools for managing the **XFS** file system

Tool Name	Purpose
`xfs_repair`	Perform consistency check and fix errors
`xfs_db`	Examine the filesystem
`xfs_info`	Display filesystem information
`xfs_admin`	Change parameters

Btrfs

`Btrfs`, short for B-tree filesystem, has built-in filesystem level compression support. The `Btrfs` filesystem keeps a checksum of the data stored. It can detect data corruption and recover the corrupted file. Instead of journaling, the `Btrfs` system employs **Copy-on-Write (COW)**. The **COW** method is an efficiency in that a copy of data is only made when a change is made - otherwise the copy is a reference to the original data. The `Btrfs` filesystem allows you to snapshot the system and save it as a read-only file. `Btrfs` has filesystem level deduplication, it will remove duplicate copies of data from the system thereby saving space.