~~30-45-MINUTE-DEEP-DIVE~~ ~1H:12M DEEP DIVE

File Managers Face-Off

Windows vs. Linux

File Explorer vs. GNOME Files · NTFS vs. ext4/XFS · GUI + Command-Line

30 SLIDES | 30-45 MIN VIDEO | GUI + CLI | 4 PLATFORMS



30-45-MINUTE-DEEP DIVE

File Managers Face-Off

Windows vs. Linux

File Explorer vs. GNOME Files · NTFS vs. ext4/XFS · GUI + Command-Line

30 SLIDES | 30-45 MIN VIDEO | GUI + CLI | 4 PLATFORMS



A 30-45-MINUTE-DEEP DIVE

File Managers Face-Off: Windows vs. Linux

File Explorer vs. GNOME Files on RHEL, Rocky Linux & Ubuntu — NTFS vs. ext4/XFS, drive letters vs. the single directory tree, and every command you need on both sides.

- 30 Slides
- GUI + CLI Side-by-Side
- Windows · RHEL · Rocky · Ubuntu



```
PS> dir C:\Users\Wahid
Documents  Downloads  Pictures
$ ls -la ~/
drwxr-xr-x  Documents  Downloads
$ _
```

WHAT WE'LL COVER

Video Roadmap



01 **Meet the File Managers**

File Explorer vs. GNOME Files (Nautilus) — GUI tour & shortcuts

02 **File Systems Under the Hood**

NTFS/ReFS vs. ext4/XFS/Btrfs, permissions, directory structure

03 **Command-Line Face-Off**

Every core file operation, side by side, on both platforms

04 **Sharing, Cloud & Disks**

Network shares, cloud integration, disk tools, recovery

05 **Wrap-Up & Cheat Sheet**

A command equivalents reference you can keep

01

MODULE 1

Meet the File Managers

Before touching a single command, let's tour the graphical file managers admins actually click through every day on both platforms.

- File Explorer
- GNOME Files (Nautilus)
- Shortcuts
- Feature Comparison

Windows File Explorer

Explorer has been Windows' file manager since Windows 95 — and it still gets meaningful updates today.

R

Ribbon / Command Bar

Context-aware actions across the top — copy, share, sort, and view options change per folder type

Q

Quick Access & Pinning

Pin frequently used folders to the sidebar for one-click access

T

Tabs (Windows 11)

Multiple folders open in one window, browser-style, instead of separate windows

O

OneDrive Integration

Cloud files show sync status icons directly in the folder view

GNOME Files (Nautilus): One App, Three Distros

RHEL, Rocky Linux, and Ubuntu all ship the GNOME desktop by default — so all three use the exact same file manager, branded simply "Files."

S

Same App Everywhere

What you learn on Ubuntu transfers directly to RHEL and Rocky — it's one codebase (Nautilus/GTK)

L

List & Grid Views

Toggle between icon grid and detailed list with file size, type, and modified date

B

Batch Rename

Select multiple files and rename them all at once with a numbering pattern

O

GNOME Online Accounts

Google Drive, Nextcloud, and other cloud accounts mount directly into the sidebar

GUI Feature Comparison

Windows File Explorer

- Tabs, Quick Access pinning, and a context-aware ribbon
- Built-in search indexing across the whole PC
- Native OneDrive sync status icons
- Built-in zip creation and extraction

GNOME Files (Nautilus)

- Tabs and bookmarks, minimal chrome, GTK-native design
- Search is fast but not pre-indexed like Windows by default
- Cloud accounts via GNOME Online Accounts, or rclone for more providers
- Built-in archive creation via the right-click menu (uses tar/zip under the hood)

Both are genuinely capable daily drivers — the differences are mostly about which ecosystem they're wired into.

Keyboard Shortcuts Cheat Sheet

File Explorer

- F2 — Rename · Delete — Move to Recycle Bin
- Shift+Delete — Permanent delete
- Ctrl+Shift+N — New folder
- Alt+Enter — Properties
- Win+E — Open File Explorer

GNOME Files (Nautilus)

- F2 — Rename · Delete — Move to Trash
- Shift+Delete — Permanent delete
- Ctrl+Shift+N — New folder
- Alt+Enter — Properties
- Super — Open Activities, then type to search

Notice how similar these are — both borrowed heavily from the same desktop-metaphor conventions.

02

MODULE 2

File Systems Under the Hood

Click past the GUI and the real differences show up: how data is actually organized on disk, how permissions work, and how the two worlds structure their directories entirely differently.

- NTFS / ReFS
- ext4 / XFS / Btrfs
- Permissions
- Directory Structure

Windows vs. Linux File System Architecture



Windows mounts each volume as its own lettered drive; Linux mounts every volume somewhere inside one single directory tree.

Permissions Models: ACLs vs. POSIX

Windows (NTFS / ReFS)

- Access Control Lists (ACLs) — a flexible list of users/groups, each with specific rights
- Rights include Read, Write, Modify, Full Control, and more granular special permissions
- Inheritance flows down from parent folders unless explicitly broken
- Managed via the Security tab in Properties, or icacls / Get-Acl in scripts

Linux (ext4 / XFS / Btrfs)

- Classic POSIX model: Owner, Group, and Others — each with Read/Write/Execute
- Extended POSIX ACLs add Windows-style per-user/group rules when the simple model isn't enough
- Every file's mode is visible instantly with ls -l (e.g. rwxr-xr-x)
- Managed with chmod, chown, chgrp, and setfacl/getfacl for ACLs

Linux's default model is simpler by design; Windows' ACLs are more granular out of the box.

Directory Structure: Drive Letters vs. a Single Tree

C:\ (system drive)	/ (root — everything starts here)
C:\Windows — OS files	/home/<user> — personal files
C:\Program Files — installed apps	/etc — system configuration
C:\Users\<name> — personal files	/var — logs, spools, variable data
C:\ProgramData — shared app data	/usr — installed applications
D:\, E:\ ... (separate drives/partitions)	/mnt, /media — other drives mount HERE, not as new letters

This is the single biggest mental shift moving between the two: Linux has no "D drive" — everything is a folder somewhere under /.

Small Differences That Trip Everyone Up

Windows

- Case-insensitive: Report.txt and report.txt are the same file
- Backslash path separator: C:\Users\Wahid\Documents
- Hidden files use a file attribute (Hidden), toggled in folder view options
- Reserved characters: < > : " / \ | ? * can't appear in file names

Linux

- Case-sensitive: Report.txt and report.txt are two different files
- Forward-slash path separator: /home/wahid/Documents
- Hidden files simply start with a dot: .bashrc, .config
- Only / and the null character are truly forbidden in file names

Case sensitivity is the #1 source of "it works on my machine" bugs when moving projects between the two.

03

MODULE 3

Command-Line Face-Off

Every core file operation, shown side by side: PowerShell and Command Prompt on Windows against bash on RHEL, Rocky, and Ubuntu.

Navigation | Copy/Move/Delete | Permissions | Search | Archives | Symlinks

Navigation & Listing

- Both shells track a 'current directory' that every relative path is measured from
- PowerShell's Get-ChildItem (aliased dir or ls) returns real objects, not just text
- Linux's ls -la shows permissions, owner, size, and hidden files in one view
- Tab-completion works in both PowerShell and bash — use it constantly

navigation

```
PS>
Get-Location
PS>
Set-Location C:\Users\Wahid\Documents
PS>
dir
$
pwd
$
cd ~/Documents
$
ls -la
```

Copy, Move, Delete & Create

- PowerShell's cmdlets are verbose but self-documenting: Copy-Item, Move-Item, Remove-Item
- Linux's classic commands are short by design — cp, mv, rm — built for fast, repeated typing
- Both support wildcards and recursive flags for whole-folder operations
- rm -rf and Remove-Item -Recurse -Force are equally destructive — always double-check the path first

file-ops

```
PS>
Copy-Item report.docx D:\Backup\
PS>
Move-Item old.log C:\Archive\
PS>
Remove-Item temp.txt
PS>
New-Item -ItemType Directory -Name Projects
$
cp report.docx ~/backup/
$
mv old.log ~/archive/
$
rm temp.txt
$
mkdir projects
```

Permissions from the Command Line

- icacls is the classic Windows CLI for ACLs; Get-Acl/Set-Acl are the PowerShell-native equivalents
- chmod changes read/write/execute bits; chown changes the owning user; chgrp changes the group
- Numeric mode (chmod 755) is faster to type once you've memorized the read=4/write=2/execute=1 pattern
- Always check current permissions before changing them — ls -l or icacls with no other arguments

permissions

```
PS>
icacls report.docx /grant Wahid:F
PS>
Get-Acl report.docx | Format-List
$
chmod 755 script.sh
$
chmod u+x script.sh
$
chown wahid:developers report.docx
$
ls -l report.docx
```

Searching for Files & Content

- Get-ChildItem -Recurse walks an entire folder tree; pipe to Where-Object to filter
- find searches by name, type, size, or modification time; locate uses a pre-built index for instant results
- grep -r searches inside file contents recursively — invaluable for finding a config value fast
- Windows' Everything (a popular third-party tool) gives Explorer instant, indexed search similar to locate

search

```
PS>
Get-ChildItem -Path C:\Projects -Recurse -Filter
*.log
PS>
Get-ChildItem -Recurse | Where-Object {$_.Length -gt
100MB}
$
find ~/projects -name "*.log"
$
locate nginx.conf
$
grep -r "api_key" ~/projects
```

Archiving & Compression

- Compress-Archive/Expand-Archive handle .zip natively in PowerShell with no extra install
- tar is the Linux workhorse — bundles files, usually paired with gzip (.tar.gz) for compression
- zip/unzip on Linux gives you Windows-compatible .zip files when that's what you need to share
- Right-click 'Compress to ZIP file' / 'Compress' both do the same thing from the GUI on either OS

archives

```
PS>
Compress-Archive -Path .\Reports -DestinationPath
reports.zip
PS>
Expand-Archive -Path reports.zip -DestinationPath
.\Restored
$
tar -czvf reports.tar.gz reports/
$
tar -xzvf reports.tar.gz
$
zip -r reports.zip reports/
$
unzip reports.zip
```

Symbolic Links & Junctions

- A symbolic link is a pointer to another file or folder — opening it transparently opens the target
- Windows needs mklink (and often admin rights); Linux's ln -s works for any user in their own space
- Junctions (Windows) link folders on the same machine without needing elevated privileges
- Both platforms use links heavily under the hood — Linux especially, for versioned libraries

symlinks

```
PS>
New-Item -ItemType SymbolicLink -Path shortcut.txt -
Target C:\real\file.txt
PS>
mklink /D C:\link-folder C:\real-folder
$
ln -s ~/real/file.txt ~/shortcut.txt
$
ln -s ~/real-folder ~/link-folder
$
ls -l shortcut.txt   # shows the -> target
```

04

MODULE 4

Sharing, Cloud & Disks

Files rarely live in isolation — here's how each platform shares folders over the network, connects to the cloud, and manages the disks underneath.

SMB / Samba

Cloud Integration

Disk Management

Recovery & Versioning

Network File Sharing: SMB Everywhere

Windows

- SMB (Server Message Block) is native — right-click a folder, choose 'Give access to'
- Mapped network drives show up as new drive letters (Z:\ etc.)
- SMB over QUIC (Windows Server 2025) allows encrypted, VPN-free remote file access

RHEL / Rocky / Ubuntu

- Samba lets Linux serve SMB shares that Windows machines browse natively
- NFS (Network File System) is the traditional Linux-to-Linux sharing protocol
- Shares mount into the directory tree (e.g. /mnt/nas) rather than getting their own drive letter

Samba means a Linux file server and a Windows PC can share folders with each other with zero extra software on the Windows side.

Cloud Integration & Mount Points

Windows

- OneDrive is built into Explorer with real-time sync status icons
- Files On-Demand keeps a placeholder locally, downloading content only when opened
- Other clouds (Google Drive, Dropbox) install their own Explorer integration

RHEL / Rocky / Ubuntu

- GNOME Online Accounts adds Google Drive/Nextcloud straight into the Files sidebar
- rclone is the power-user's Swiss Army knife — scriptable sync with 70+ cloud providers
- Any mounted cloud or network folder just appears as another path under / — no new drive letter

The underlying theme repeats: Windows adds a new letter, Linux adds a new folder.

Disk Management Tools

Windows

- Disk Management (diskmgmt.msc) — GUI for partitions, formatting, and drive letters
- diskpart — scriptable command-line equivalent for the same tasks
- Get-Disk / Get-Volume — modern PowerShell cmdlets for inventory and automation

RHEL / Rocky / Ubuntu

- GParted — the graphical partition editor most Linux desktops reach for
- lsblk / fdisk / parted — command-line tools for listing and partitioning disks
- df -h and du -sh give quick space-used summaries the GUI often hides

Command-line disk tools are non-negotiable knowledge on Linux servers — there's usually no GUI to fall back on.

Recovery & Versioning

Windows

- Recycle Bin catches everyday accidental deletes — right-click Restore
- File History / Previous Versions restores an earlier copy from a Shadow Copy snapshot
- System Restore rolls back system-wide changes, not just individual files

RHEL / Rocky / Ubuntu

- Trash in GNOME Files works the same way as the Recycle Bin for GUI deletes
- Deletes from the terminal (rm) skip the Trash entirely — there's no undo
- Btrfs/LVM snapshots and tools like Timeshift give point-in-time system rollback

The biggest gotcha for new Linux admins: terminal deletes are usually permanent, immediately.

05

MODULE 5

Wrap-Up & Cheat Sheet

Let's bring it all together with a quick-reference command table and the big-picture takeaways for admins who work across both worlds.

Command Equivalents

Common Tasks

Key Takeaways

Command Equivalents Cheat Sheet



Windows (PowerShell)	Linux (bash)
Get-ChildItem (dir)	ls
Set-Location (cd)	cd
Copy-Item / Move-Item / Remove-Item	cp / mv / rm
Get-Content (type)	cat
Select-String (grep-like)	grep
Get-Acl / icacls	ls -l / chmod / chown

Screenshot this slide — it's the fastest way to translate muscle memory between the two shells.

Common Admin Tasks: Quick Reference

Windows

- Show hidden files: View > Show > Hidden items
- Batch rename: select files, F2, type a name (auto-numbers the rest)
- File properties: Alt+Enter or right-click > Properties
- Check disk space: This PC, or Get-PSDrive in PowerShell

RHEL / Rocky / Ubuntu

- Show hidden files: Ctrl+H in GNOME Files, or ls -a in the terminal
- Batch rename: select files, F2 in Nautilus, or a for loop in bash
- File properties: right-click > Properties, or stat filename in the terminal
- Check disk space: df -h (volumes) or du -sh (folder sizes)

Key Takeaways for Admins Working Across Both Worlds



1

One Tree vs. Many Drives

Linux's single directory tree is the biggest conceptual shift for Windows admins

2

Case Sensitivity Bites

Assume Linux paths and filenames are case-sensitive until proven otherwise

3

Terminal Deletes Are Forever

rm has no Recycle Bin — always double-check the path before hitting Enter

4

Same App, Three Distros

Learning GNOME Files once means you already know it on RHEL, Rocky, and Ubuntu

You Can Now Navigate Either World with Confidence

Same fundamental job — organizing files — approached with two very different philosophies. Now you speak both.

Practice	Try every command shown here in a lab VM on each platform back to back.
Go Deeper	This pairs perfectly with the full RHEL, Rocky, Ubuntu, and Windows Server courses.
Bookmark It	Keep the cheat-sheet slide handy for the next time you switch platforms mid-task.
Thank You	Questions and feedback are always welcome — happy administering!

Enjoyed This Video? Let's Stay Connected

Free Linux and Windows training — new videos every week.

Subscribe	@MyWebUniversity (English) and @MyWebUniversity-Dari (Dari) — 2,600+ free videos and counting
Comment	Tell us which platform comparison you want to see next
Share	Send this to a teammate who's about to switch between Windows and Linux
Read & Buy the Book	Free chapters, the full 43-chapter Linux book, and every link — all on MyWebUniversity.com