rsync trailing slash: what it actually copies

📅 2026-09-22 📊 beginner ["rsync" "linux" "backup"]

The Problem

You run `rsync -av /var/log /backup/` expecting `/backup/log/` to appear, but half your scripts fail because you got `/backup/var/log/` instead. Or the opposite: you expected the source folder inside the destination and you got a flat copy of its contents.

The Mistake

Treating `source` and `source/` as the same argument. They're not. The trailing slash tells rsync "copy the contents of this directory", not "copy this directory."

The Fix

```
# Copy the CONTENTS of /var/log into /backup/log/
rsync -av /var/log/ /backup/log/

# Copy the FOLDER /var/log into /backup/ (so you get /backup/log/)
rsync -av /var/log /backup/
```

Why It Works

rsync uses the trailing slash as a "treat as directory contents" signal:

- `rsync src/ dst/` → for every entry inside `src`, copy it into `dst`.
- `rsync src dst/` → copy `src` (a directory) into `dst`, creating `dst/src`.

This is the single most common rsync footgun. The `man rsync` page calls it out in the first paragraph; nobody reads it.

Copy-Paste

```
# Mirror /home to /backup, keeping folder structure
rsync -av --delete /home/ /backup/home/
```

Next Steps

- Add `-n` (dry-run) to any rsync command the first time you run it.
- Add `--itemize-changes` to see exactly what rsync is doing per file.
- For backups over SSH: `rsync -av -e ssh src/ user@host:/dst/`.

Download PDF (1 page)