The importance of sudo (rsync)

Adding `sudo` prevents "permission denied" errors if the blockage is caused by a lack of read or write privileges on files and folders. However, how you apply `sudo` depends on whether you are running `rsync` locally or over a remote SSH connection. [1, 2, 3]

Local Transfers

Put `sudo` at the very front of your command: `sudo rsync -av /source/ /destination/`.
This runs the entire process with root permissions, bypassing local file restrictions. [1, 2, 3]

Remote Transfers (via SSH)

Typing `sudo rsync` locally only elevates privileges on your local machine. If the permission error happens on the remote server, you must tell `rsync` to invoke `sudo` on the remote side using the `--rsync-path` flag: [1, 2, 3, 4]

Command: `rsync -av -e ssh --rsync-path="sudo rsync" /local/path/ user@remote:/remote/path/`
Important: The remote user account must be configured for passwordless `sudo` (`NOPASSWD`) for `rsync` in the remote `/etc/sudoers` file. Otherwise, `rsync` will hang or fail when it tries to prompt you interactively for a password.

Revision #1
Created 2026-08-10 15:07:35 UTC by Admin
Updated 2026-08-10 15:08:47 UTC by Admin