Transfer files to a Linux VPS safely with SFTP

SFTP transfers files through SSH. It does not make a live application safe to overwrite. This procedure suits Debian and Ubuntu VPS accounts with SSH file access and a writable staging directory. You do not need root simply to transfer files you own. Application databases and changing game worlds need their own consistent export or shutdown procedure.

Check both ends before uploading

Locate the exact local files and confirm that your account can read them. On the server, check the intended directory with `pwd`, `ls -ld` and `df -h .`. Leave room for both the old version and the upload. A successful connection to a similarly named machine is not proof that you selected the correct service.

Use the hostname and SSH port from your service details. On a first connection, verify the host-key fingerprint through a trusted channel. An unexpected changed fingerprint requires investigation; deleting the saved host key merely hides the warning.

```
sftp -P 22 username@server.example.com
pwd
lpwd
ls
```

The uppercase `-P` selects the SSH port. The SFTP prompt has its own commands; `lpwd` shows the local directory. Replace the example account, hostname and port. Never paste a password into a command or support message.

Upload a separate copy

Create a staging directory you own, outside any public web root when the files include configuration or private data. In the SFTP session, change into that directory, then upload a clearly named file. Quote paths containing spaces.

```
cd uploads-staging
put "release.zip" "release-new.zip"
ls -l
bye
```

A disconnected transfer may leave a partial destination. Do not rename that partial file into production. Resume only when you know the local source and remote partial content belong to the same unchanged transfer; otherwise upload under another name. The SFTP manual explains why resuming mismatched content can produce corruption.

Verify before using the upload

Compare the byte count and SHA-256 digest on both ends with the appropriate local checksum tool and `sha256sum release-new.zip` on Linux. An archive listing checks readability, but does not prove that application data was captured consistently. Keep the original source until the destination has passed these checks.

Extract into a separate release directory rather than over the running installation. Check archive paths before extraction, review ownership and permissions, and keep secrets outside publicly served folders. A transfer does not need globally writable permissions. If an application account cannot read a file, diagnose that account and path rather than using recursive permission changes.

Recover from an incomplete transfer

Leave the working application untouched. Correct the connection or available-space issue and repeat the upload to staging. Switch the application only after its normal deployment checks pass, keeping the previous release for rollback. When asking for help, provide the transfer time, client error, destination filesystem and redacted path. Do not attach private keys, passwords or configuration archives containing secrets.

Related instructions

Connect using SSH. Verify a downloaded backup.

Reference: Official documentation. Check the documentation installed on your own operating-system release when an option differs.

Revision #1
Created 2026-11-09 10:42:42 UTC by CloudExa Hosting Admin
Updated 2026-11-09 10:42:42 UTC by CloudExa Hosting Admin