



[What is Cloud Computing?](#) › [Cloud Computing Concepts Hub](#) › [Migration](#)

What is Secure File Transfer Protocol (SFTP)?

Create an AWS account

What is Secure File Transfer Protocol?

Why is the Secure File Transfer Protocol important?

What is the S in SFTP?

How does Secure File Transfer Protocol work?

What are some common SFTP commands?

How does SFTP compare to other file transfer protocols?

What are some best practices in using SFTP?

How can AWS support your file storage requirements?

What is Secure File Transfer Protocol?

Secure File Transfer Protocol (SFTP) is a standard networking protocol for the secure transfer of files between connected systems. A network protocol is a set of established rules that act as a common language regardless of the specific hardware or software used by the devices on the network. SFTP adds authentication and encryption to file transfer, allowing organizations to share files securely and meet regulatory compliance requirements for sensitive file data.

Why is the Secure File Transfer Protocol important?

Data security and confidentiality are basic requirements for file transfer over any network, whether within an enterprise local area network (LAN) or over the Internet. The Secure File Transfer Protocol is used alongside other common network security protocols, such as [HTTPS](#) (Hypertext Transfer Protocol Secure), to ensure file data security at the network level.

SFTP was developed to replace File Transfer Protocol (FTP), a traditional method of file transfer. It is used for transferring sensitive files in many of the world's most widely used apps, such as Git, [VSCode](#), Ansible, WordPress, and AWS Transfer Family.

What is the S in SFTP?

SFTP is alternatively known as SSH File Transfer Protocol, due to its use of the Secure Shell (SSH) connection protocol. S can stand for both Secure and Secure Shell.

SSH

The SSH protocol is an authentication-based cryptographic connection protocol for network communications. SSH handles secure data transport, authentication, and connections over TCP (Transmission Control Protocol). TCP is one of the main internet transport protocols.

SSH ensures that any listener between two parties on a network cannot decipher any shared, confidential communications once a session has been established. For instance, a listener on a network would not be able to view passwords or Personally Identifiable Information (PII) sent over a connection with Secure Shell encryption.

SSH gives the client (user) permission to execute basic commands on a remote server; however, SSH lacks the ability to transfer and manipulate files. For secure data transfer of files and file manipulation operations, SSH requires a layered approach, in combination with SFTP.

How does Secure File Transfer Protocol work?

SFTP allows users to send, receive, and manipulate server files and folders, reducing the risk of leaked information due to a compromised connection. SFTP encrypts all file contents and commands, making them indecipherable in transit, for assurance in file transfer compliance.

Client-server architecture

The SFTP protocol uses the SSH client-server architecture model for communication. This architecture model gives the server control over connections, resourcing, and security, even though the client initiates a session.

Public and private key infrastructure

SFTP uses public and private SSH keys to authenticate sessions between clients and servers. Using SSH key infrastructure ensures minimal security management overhead, which only involves the SFTP client and SFTP server.

Port 22 by default

SFTP utilizes port 22 by default for reliable data delivery with SSH connections over TCP. However, it's possible to name a different port in SFTP, in the case of existing connections on port 22.

Resumption of interrupted transfers

SFTP allows users to resume file transfers that were previously paused due to interrupted sessions. Resuming file transfers heightens productivity, even with unstable connections.

What are some common SFTP commands?

Users can perform SFTP commands via the command line interface (CLI) or by following click-through workflows in an app with built-in SFTP. Windows, Mac, and Linux users all have access to the SFTP command line via a shell app. Here are some useful SFTP commands using the bash shell.

Initiating and closing an SFTP session

`sftp user@hostname` e.g. `sftp dan@example.com`

Opens a new SFTP connection on the example.com server. The command prompt in the shell will change to **sftp>** to indicate the session has commenced. There are various options that you can include in this command, such as:

- `-P [number]` to specify a port number
- `-i [file]` to include a private key file, and
- `-r` to switch on recursive directory transfer.

sftp exit closes an SFTP session.

Transferring files

get server_path_and_filename local_path copies the given file from the server to the specified directory.

e.g. `get /projects/marketing/proposal.txt /newopportunities`

put local_path_and_filename server_path transfers a local file to the given server directory.

e.g. `put /finances/invoices/march.txt /projects/marketing/payments`

Similarly, the **reget** and **reput** commands resume interrupted transfers.

Remote file management

- **chown user path** changes the ownership of the file or folder at the given path on the server to the specified user.
- **chmod number path** changes the permissions of the file or folder at the given path on the server.
- **ls** shows the list of files and folders in the current server directory.
- **cd path** navigates to the given directory on the server.
- **mkdir dir_name** creates a new folder on the server.
- **rmdir dir_name** removes a given folder on the server.
- **rename old_file_name new_file_name** renames a given file on the server.
- **pwd** shows the current directory on the server.
- **lpwd** shows the current local directory.

How does SFTP compare to other file transfer protocols?

SFTP is one of a range of standard file transfer protocols.

SFTP vs. FTPS

FTP Secure (FTPS) uses the Transport Layer Security (TLS) protocol and Secure Socket Layer (SSL) protocol over the top of the File Transfer Protocol to conduct secure file transfers. Authentication with FTPS relies on [SSL certificate](#) management practices. FTPS is considered mainly deprecated.

FTPS builds on FTP, whereas SFTP was introduced to replace FTP with a more secure architecture. SFTP is used much more widely than FTPS today, due to its simplicity, security, and cloud support.

SFTP vs. TFTP

Trivial File Transfer Protocol (TFTP) is a lightweight protocol for transferring files that utilizes a client-server architecture. TFTP does not contain any built-in security measures and so must be layered with security protocols or protections for file transfers in regular environments.

While SFTP is used in a wide variety of applications, TFTP is typically used only directly on a local, secure network when setting up or booting new hardware, firmware, and operating systems.

What are some best practices in using SFTP?

Use best practices for public-private key management, including key rotation and secure storage on the SFTP server. Ensure your SFTP server is always up to date with security updates and patches, plus uses strong encryption such as AES. Log successful file transfers and failed access attempts for anomaly detection and response.

Remember that SFTP is just one part of network security. Firewalls, intrusion detection systems, and other security measures should be tailored to your specific network architecture.

How can AWS support your file storage requirements?

Amazon offers a range of secure file storage and secure file transfer solutions.

[Amazon S3](#) is object storage built to store and retrieve any amount of data from anywhere. You can use it to store and retrieve any amount of data, including files, at any time, from anywhere, including automated file transfers.

[Amazon Elastic File System \(EFS\)](#) provides serverless, fully elastic cloud file storage that enables you to create file systems accessible to [Amazon Elastic Compute Cloud \(EC2\)](#) instances and other AWS cloud services. It supports full file system access semantics, such as strong consistency and file locking. You can share file data without provisioning or managing storage capacity and performance.

[AWS Transfer Family](#) provides fully managed support to transfer files securely over SFTP, AS2, FTPS, FTP, and web browsers directly into and out of Amazon S3 or Amazon EFS. You can seamlessly migrate, automate, and monitor your file transfer workflows by maintaining existing client-side configurations for authentication, access, and firewalls, so nothing changes for your customers, partners, and internal teams, or their applications.

Get started with file storage on AWS by [creating a free account](#) today.

Next steps on AWS

Check out additional product-related resources



Sign up for a free account

Instantly get access to the AWS Free Tier.



Start building in the console

Get started building with AWS in the AWS Management Console.



Create an AWS account

English

Learn

What Is AWS?

What Is Cloud Computing?

What Is Agentic AI?

Cloud Computing Concepts Hub

AWS Cloud Security

What's New

Resources

Getting Started

Training

AWS Trust Center

AWS Solutions Library

Architecture Center

Product and Technical FAQs

Developers

Builder Center

SDKs & Tools

.NET on AWS

Python on AWS

Java on AWS

PHP on AWS

Help

Contact Us

File a Support Ticket

AWS re:Post

Knowledge Center

AWS Support Overview

Get Expert Help

[Blogs](#)

[Analyst Reports](#)

[JavaScript on AWS](#)

[Press Releases](#)

[AWS Partners](#)



Hi, I can connect you with an AWS representative or answer questions you have on AWS.



[Back to top](#) ↑



Need more info? Highlight any text to get an explanation generated with AWS generative AI.



Amazon is an Equal Opportunity Employer: Minority / Women / Disability / Veteran / Gender Identity / Sexual Orientation / Age.



[Privacy](#)

[Site terms](#)

[Cookie Preferences](#)

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.



2