How to Use SFTP Commands and Options

December 1, 2021

COMMANDS SFTP

Contents

Introduction

SFTP (Safe File Transfer Protocol) is part of the SSH protocol designed to securely transfer files between remote systems. It allows users to view, manage, and change file and directory permissions on remote systems.

In this tutorial, we will go over the commands you can use with SFTP while providing explanations, options, and examples for each.

Prerequisites

- Access to a local system and a remote server, connected using an SSH public key pair.
- A working Internet connection.
- Access to the terminal window.

SFTP Commands and Options List

SFTP allows users to transfer data between a remote SFTP server and a local client system. SFTP uses the SSH network protocol to connect two systems that share a public SSH key.

Connecting to the SFTP server opens the SFTP shell interface. The SFTP shell interface supports the following commands:

Command	Description
`cd [path]`	Change the directory on the remote server to `[path]`.
`lcd [path]`	Change the directory on the local system to `[path]`.
`chgrp [group ID] [path]`	Change group ownership to `[group ID]` for the file or folder located at `[path]`.
`chmod [mode] [path]`	Change ownership to `[mode]` for the file or folder located at `[path]`.

help	Display the help text.
get [remote path] [local path]	Transfer a file or directory from **[remote path]** on the remote server to **[local path]** on the local system.
lls [options] [path]	Display the listing for the directory located at **[path]** on the local system. Uses the **ls** command options.
ln [old path] [new path]	Create a symlink from **[old path]** to **[new path]** on the remote server.
lmkdir [path]	Create a directory at **[path]** on the local system.
lpwd	Display the current local directory.
ls [options] [path]	Display the listing for the directory located at **[path]** on the remote server. Uses the **ls** command options.
lumask [mask]	Set local permissions mask to **[mask]**.
mkdir [path]	Create a directory at **[path]** on the remote server.
put [local path] [remote path]	Transfer a file or directory from **[local path]** on the local system to **[remote path]** on the remote server.
pwd	Display the current remote directory.
exit	Exit the SFTP interface.
quit	Exit the SFTP interface.
rename [old path] [new path]	Rename a file on the remote server from **[old path]** to **[new path]**.
rmdir [path]	Remove a directory located at **[path]** on the remote server.
rm [path]	Remove a file located at **[path]** on the remote server.
symlink [old path] [new path]	Create a symlink from **[old path]** to **[new path]** on the remote server.
version	Display the current version of SFTP

!	Temporarily move to the local shell.
?	Display the help text.

Connecting to SFTP

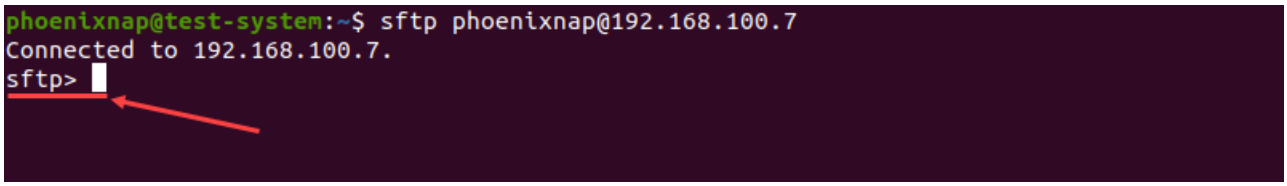
Connecting to SFTP uses the same syntax as connecting to a remote system with SSH:

```
sftp [username]@[remote hostname or IP address]
```

For instance, connecting to a server with the *phoenixnap* username at the IP address *192.168.100.7*:

```
sftp phoenixnap@192.168.100.7
```

If the connection is successful, the shell moves to the SFTP interface, indicated by **sftp>** in place of the current username:

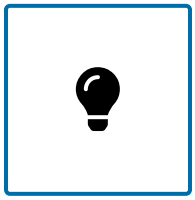


When connecting to a remote system with SFTP, use the following options with the **sftp** command to change its behavior:

Option	Description
-1	Use version 1 of the SSH protocol when connecting.
-4	Use IPv4 addresses only.
-6	Use IPv6 addresses only.
-A	Allows the forwarding of SSH authentication agent to the remote server.
-a	Attempt to continue interrupted file transfers.

`-b [batch file]`	Specify a batch file to start the `sftp` command in batch mode.
`-C`	Use file compression.
`-c [cipher]`	Select a cipher to use when encrypting data for transfer.
`-D [SFTP server path]`	Connect to a local SFTP server without using SSH.
`-F [SSH configuration file]`	Specify an SSH configuration file to use when connecting.
`-f`	Flush files to disk immediately after transfer.
`-i [private key file]`	Select a file that contains the private key for public key authentication.
`-J [destination]`	Set up TCP forwarding via the destination provided.
`-l [kbit/s]`	Set a limit to the connection bandwidth in kbit/s.
`-N`	Disable quiet mode.
`-o [SSH option]`	Add an ssh command option when connecting to SFTP.
`-P [port number]`	Set a port to connect to.
`-p`	Preserve file permissions and access times when transferring.
`-q`	Enable quiet mode.
`-R [number of requests]`	Set the number of allowed concurrent file transfer requests.
`-r`	Transfer directories recursively.
`-S [client]`	Specify an SFTP client you want to use to connect.
`-s [SSH subsystem or SFTP server path]`	Select an SSH2 subsystem or SFTP server path.
`-v`	Keep a verbose session log.

Use the **`exit`** command to end the current connection:



Note: Learn everything you need to know about how SSH works in our article How Does SSH Work?.

Transferring Files

Use the **get** and **put** commands to create a file transfer request in SFTP. The **get** command transfers the files from a remote server to the local system, while the **put** command does the opposite.

The **get** command uses the following basic syntax:

```
get [path to file]
```

Using the **get** command transfers a file from the remote server to the local system's *Home* directory. For instance:

```
get example01.txt
```

On the other hand, using the **put** command transfers a file from the local system to the remote server's *Home* directory:

```
put example02.txt
```

To transfer the file to a different directory, append the name of the directory to the end of the **get** or **put** command:

```
get [path to file] [path to directory]
put [path to file] [path to directory]
```

To change the name of the file on the local system, append the new filename to the end of the command:

```
get [path to file] [new file name]
put [path to file] [new file name]
```

The **get** and **put** commands use the following options:

Option	Description
-a	Attempt to resume a file transfer.
-f	Flush the file to disk immediately after transfer.
-p	Preserve file permissions and access times while transferring.
-R	Transfer an entire directory recursively. When using this option, define a path to a directory instead of a path to a file.

Note: Learn more in our guide on how to transfer files using SFTP.

hown command changes file ownership for individual users:

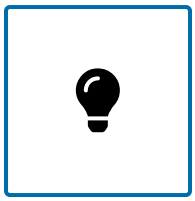
```
chown [user ID] [path to file]
```

Unlike the **chown** command, which requires a user ID, the chmod command works the same as in the standard shell:

```
chmod [permission] [path to file]
```

Another option is to use the **chgrp** command to change the group ownership of a file:

```
chgrp [group ID] [path to file]
```



Note: Learn how you can view user and group ID for the remote server.

SFTP also lets you set up a local umask, changing the default permission for all future files transferred to the local system. Use the **lumask** command to set up a new local umask:

```
lumask [permission mask]
```

Managing Files and Directories

SFTP provides options that allow users to review and manage files on both the local system and remote server. The **ls** command lets you list out the files and directories on the remote server. For instance:

Similarly, the **lls** (local **ls**) command lists files and directories on the local system:

```
lls -l
```

Note: Both the **ls** and **lls** command in SFTP use standard **ls** command options. Learn more in our guide to the Linux ls command.

The **cd** and **lcd** commands change the current working directory on the remote server or local system, respectively:

```
cd [path to directory on the remote server]
lcd [path to directory on the local system]
```

```
mkdir [path to the new directory on the remote server]
```

For instance, creating *Example_Directory* in the *Home* directory:

```
mkdir Example_Directory
```

The **`mkdir`** command has no output, so you need to use the **`ls`** command to verify the result:

Similar to this, the **`lmkdir`** command creates a directory on the local system:

```
lmkdir [path to the new directory on the local system]
```

Using the same example:

```
lmkdir Example_Directory
```

The **rename** command changes the name of a file or directory on the remote server:

```
rename [old path] [new path]
```

For example, renaming *example01.txt* to *sampledoc.txt*:

```
rename example01.txt sampledoc.txt
```

Using the **rm** command removes a file from the remote server:

For instance, removing the *sampledoc.txt* file:

```
rm sampledoc.txt
```

Similarly, the **rmdir** command removes a directory from the remote server:

```
rmdir [path to directory]
```

For example, removing *Example_Directory*:

```
rmdir Example_Directory
```

Note: Learn more in our guide to removing files and directories in Linux.

The **ln** and **symlink** commands create a symbolic link to a file or directory on the remote server:

```
ln [old path] [new path]
symlink [old path] [new path]
```

For instance, creating a link to *example02.txt* named *example_link* using the **ln** command:

```
ln example02.txt example_link
```

The **pwd** command shows the current working directory on the remote server as the output:

```
pwd
```

On the other hand, the **lpwd** command creates an output that shows the current working directory on the local system:

```
lpwd
```

Running Local Shell Commands

SFTP allows you to run a command using the local shell by adding an exclamation mark (**!**) before the command. This lets users run commands that aren't a part of the standard SFTP shell on the local system.

For instance, SFTP does not support the **tree** command in Linux. By using the local shell, you can run this command in the SFTP interface:

Using the exclamation mark (**!**) without a command temporarily moves the user to the local shell. To return to the SFTP shell, use the **`exit`** command.

Other Commands

Using the **`help`** or **`?`** commands displays the help text for the SFTP interface. The help provides a list of commands available in the SFTP shell.

The **`version`** command displays the current version of the SFTP protocol installed:

SFTP Cheat Sheet PDF

Below you can find a one-page reference sheet containing all the SFTP commands and options mentioned above. Save it as a PDF file by clicking the link below.

DOWNLOAD SFTP Cheat Sheet

Aleksandar Kovačević

With a background in both design and writing, Aleksandar Kovacevic aims to bring a fresh perspective to writing for IT, making complicated concepts easy to understand and approach.

Next you should read

SysAdmin, Web Servers

How Does SFTP Work?

November 18, 2021

SFTP is a component of SSH that allows users to safely transfer

A ou should have a solid understanding of SFTP commands and t e able to connect to a remote server with SFTP and use

s this article helpful? Yes No

us
in
g
th
e

Networking, SysAdmin, Web Servers

What is SSH?

September 22, 2021

The SSH protocol offers system administrators a way to establish a secure connection that will protect them against malicious cyber-attacks, such as password-sniffing.

READ MORE

Networking

How Does SSH Work?

December 17, 2020

SSH is a network protocol designed to provide a more secure network communication by employing encryption. This article deals with

ec
h
a

Networking, Security

Telnet vs. SSH: How Is SSH Different From Telnet?

May 20, 2021

Telnet and SSH are network protocols used to connect to remote system. This tutorial covers the ways they work, what makes them different, and when to use them.

READ MORE

ye
rs,
a
n
d
lis
ts
s
o
m
e
of
it
s