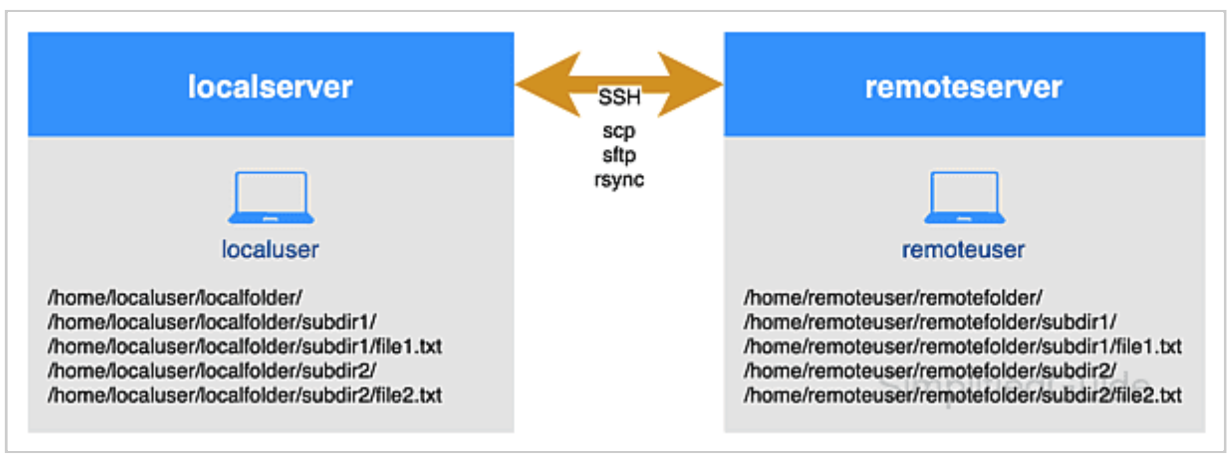
How to copy file remotely via SSH

author: shakir

Share!

`SSH` or `Secure Shell` is a protocol that allows a secure way to access remote computer. `SSH` implementation comes with `scp` utility for remote file transfer that utilises `SSH` protocol. Other applications such as `sftp` and `rsync` can also make use of `SSH` to secure its network transaction.

All these applications allow us to copy our files from local to remote server and to copy files from remote server to our local machine. Below are examples on how to use these applications for files transfers based on this setup:


localserver
SSH
scp
sftp
rsync
remoteserver
localuser
/home/localuser/localfolder/
/home/localuser/localfolder/subdir1/
/home/localuser/localfolder/subdir1/file1.txt
/home/localuser/localfolder/subdir2/
/home/localuser/localfolder/subdir2/file2.txt
remoteuser
/home/remoteuser/remotefolder/
/home/remoteuser/remotefolder/subdir1/
/home/remoteuser/remotefolder/subdir1/file1.txt
/home/remoteuser/remotefolder/subdir2/
/home/remoteuser/remotefolder/subdir2/file2.txt

Make sure you have access right to the remote server and correct permission to the remote files and folders

Method 1: Transfer file using scp

The easiest of these are `scp` or `secure copy`. While `cp` is for copying local files, `scp` is for remote file transfer. The main difference is that with `scp` you'll have to specify the remote host's `DNS (Domain Name System)` name or `IP address` and provide login credential for the command to work.

1. Copy single file from local to remote.

```
$ scp myfile.txt [email protected]:/remote/folder/
```

2. Copy single file from remote to local.

```
$ scp [email protected]:/remote/folder/myfile.txt  myfile.txt
```

3. Copy multiple files from local to remote.

```
$ scp myfile.txt myfile2.txt [email protected]:/remote/folder/
```

4. Copy all files from local to remote.

```
$ scp * [email protected]:/remote/folder/
```

5. Copy all files and folders recursively from local to remote.

```
$ scp -r * [email protected]:/remote/folder/
```

`remoteuser` need to exist and have write permission to `/remote/folder/` in the remote system.

Method 2: Transfer file using sftp

`sftp` or `Secure FTP (File Transfer Protocol)` in the other hand works almost exactly like `ftp` but with secure connection. Most of the commands are similar and can be used interchangeably. The following `sftp` example will work exactly as `ftp` would.

```
$ sftp [email protected]
Connected to 192.168.1.10.
sftp> dir
file1      file2  file3
sftp> pwd
Remote working directory: /home/user
sftp> get file2
Fetching /home/user/file2 to file2
/home/user/file2
100% 3740KB 747.9KB/s   00:05
sftp> bye
$
```

Method 3: Transfer file using rsync

You can also use `ssh` to secure your `rsync` session. To do this, use `--rsh=ssh` or `-e “ssh”` with your normal `rsync` commands. The following 2 commands will work exactly the same;

```
$ rsync -av --delete --rsh=ssh /path/to/source
[email protected]:/remote/folder/
$ rsync -av --delete -e "ssh" /path/to/source
[email protected]:/remote/folder/
```

If these options are not specified, `rsync` will first try to connect to `rsyncd` but will automatically fallback to `SSH` if `rsyncd` is not running in the remote system.

Read more guides on Secure Shell (SSH)

Discuss the article:

Comment anonymously. Login not required.

Back to top