[golinuxcloud.com](https://www.golinuxcloud.com/commands-copy-file-from-one-server-to-another-linux-unix/)

**5 commands to copy file from one server to another in Linux or Unix | GoLinuxCloud**

11-14 minutes

* [Using SFTP to copy file from one server to another](about:reader?url=https%3A%2F%2Fwww.golinuxcloud.com%2Fcommands-copy-file-from-one-server-to-another-linux-unix%2F#Using_SFTP_to_copy_file_from_one_server_to_another)
* [Using RSYNC to copy file from one server to another](about:reader?url=https%3A%2F%2Fwww.golinuxcloud.com%2Fcommands-copy-file-from-one-server-to-another-linux-unix%2F#Using_RSYNC_to_copy_file_from_one_server_to_another)
* [Using SCP to copy file from one server to another](about:reader?url=https%3A%2F%2Fwww.golinuxcloud.com%2Fcommands-copy-file-from-one-server-to-another-linux-unix%2F#Using_SCP_to_copy_file_from_one_server_to_another)
* [Using NFS to share file from one server to another](about:reader?url=https%3A%2F%2Fwww.golinuxcloud.com%2Fcommands-copy-file-from-one-server-to-another-linux-unix%2F#Using_NFS_to_share_file_from_one_server_to_another)
* [Using SSHFS to copy file from one server to another](about:reader?url=https%3A%2F%2Fwww.golinuxcloud.com%2Fcommands-copy-file-from-one-server-to-another-linux-unix%2F#Using_SSHFS_to_copy_file_from_one_server_to_another)
  + [Drawbacks of using SSHFS](about:reader?url=https%3A%2F%2Fwww.golinuxcloud.com%2Fcommands-copy-file-from-one-server-to-another-linux-unix%2F#Drawbacks_of_using_SSHFS)

In my last article I shared the [**steps to encrypt a file using gpg key**](https://www.golinuxcloud.com/tutorial-encrypt-decrypt-sign-file-gpg-key-linux/) in Linux. Now in this article I will share various commands and tools which you can use to securely copy file from one server to another in Linux. There are additionally other ways to transfer files which I cannot cover here for example you can also use HTTPS to upload and download files.

I may write another article with detail list of [steps to use HTTPS and curl for secure file upload and download](https://www.golinuxcloud.com/securely-transfer-files-https-curl-ssl-linux/).If you wish [to copy files between Windows and Linux then you can always use Samba](https://www.golinuxhub.com/2012/08/creating-samba-share.html) but since here we are targeting file transfer between two Linux machines, I will not share any steps related to Samba configuration.



**Some more articles on related topic you may be interested in**

* [*2 commands to copy folder from local to remote server or vice versa in Linux with examples*](https://www.golinuxcloud.com/ssh-copy-folder-local-to-remote-server-linux/)
* [*Step-by-Step Guide to setup SFTP chroot Jail to restrict user to a specfic directory when copying files in Linux*](https://www.golinuxcloud.com/sftp-chroot-restrict-user-specific-directory/)
* [*How to securely transfer files between two hosts using HTTPS in Linux*](https://www.golinuxcloud.com/securely-transfer-files-https-curl-ssl-linux/)

In computing, the **SSH** File Transfer Protocol (also Secure File Transfer Protocol, or **SFTP**) is a network protocol that provides file access, file transfer, and file management over any reliable data stream.

SFTP is easy to work with: You enter sftp along with the name of the remote system on the command line. You are prompted for the account password; then you are dropped into SFTP with the connection open and waiting.

You can also [automate the file transfer using SFTP in shell script](https://www.golinuxcloud.com/automate-sftp-shell-script-with-password-unix/), or you can also use [one liner SFTP commands to perform file transfer rather than interactive sessions](https://www.golinuxcloud.com/single-line-unix-linux-sftp-commands-example/).

[deepak@Ban17-inst01-a ~]$ sftp -q -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no deepak@10.43.138.2

Password:

Once you give the [password of deepak user](https://www.golinuxcloud.com/check-last-password-change-expiration-linux/) on target node, you will get sftp shell

sftp> ls -l

-rw-r----- 1 root root 401 Dec 7 11:53 new\_key.pub

-rw-r----- 1 deepak users 9 Dec 10 14:13 pwd.txt

drwxr-xr-x 2 root root 4096 Nov 28 11:38 scripts

-rw-r--r-- 1 root root 0 Dec 10 14:07 test\_file

Next to copy file from host to client (i.e. upload a file from host to client)  
Here I have a file 'pwd.txt' on my host server under '/home/deepak/pwd.txt' which I wish to copy to my client's current working directory

sftp> put /home/deepak/pwd.txt .

Uploading /home/deepak/pwd.txt to /home/deepak/./pwd.txt

/home/deepak/pwd.txt 100% 9 26.6KB/s 00:00

To copy a directory and all it's content use (-r). Here /home/deepak/mydir is available on my host machine which I am copying to the connected client node under current working directory.

sftp> put -r /home/deepak/mydir .

Entering /home/deepak/mydir/

So the file was successfully uploaded. You can verify the same

sftp> ls

new\_key.pub pwd.txt scripts test\_file

Next copy a file from client node to your host server. I have a file 'test\_file' on my client node under '/home/deepak/test\_file'

sftp> get test\_file /tmp/

Fetching /home/deepak/test\_file to /tmp/test\_file

Validate the same on your host server

# ls -l /tmp/test\_file

-rw-r----- 1 deepak deepak 0 Dec 10 14:09 /tmp/test\_file

You can get more supported options from the man page of sftp.

**Using RSYNC to copy file from one server to another**

**rsync** is a utility that you can use to copy file from one server to another very easily, and there are many options available to allow you to be very specific about how you want the data transferred. Another aspect that makes rsync flexible is the many ways you can manipulate the source and target directories. However, you don't even have to use the network; you can even copy data from one directory to another on the same server.

Copying a file within the same server from one location to another

# rsync -r /home/deepak/mydir/test /tmp/

Here we are using -r to copy recursively, you can also use (-a) i.e. for archive which retains as much metadata as possible (in most cases, it should make everything an exact copy).

NOTE:  
This works because whenever rsync runs, it will copy what's different from the last time it ran. The files from our first backup were already there, but the permissions were wrong. When we ran the second command, rsync only needed to copy what was different, so it applied the correct permissions to the files. If any new files were added to the source directory since we last ran the command, the new or updated files would be copied over as well.

To copy files between two servers

# rsync -av test deepak@10.43.138.2:/tmp/

Password:

sending incremental file list

sent 44 bytes received 12 bytes 22.40 bytes/sec

total size is 5 speedup is 0.09

**Using SCP to copy file from one server to another**

A useful alternative to rsync is the Secure Copy (**SCP**) utility to copy file from one server to another, which comes bundled with OpenSSH. It allows you to quickly copy files from one node to another. If your goal is to send a single file or a small number of files to another machine, SCP is a great tool you can use to get the job done. To utilize SCP, we'll use the scp command. Since you most likely already have OpenSSH installed, you should already have the scp command available

Using SCP is very similar in nature to rsync. The command requires a source, a target, and a filename. To transfer a single file from your local machine to another, the resulting command would look similar to the following:

# scp -q -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null host\_list deepak@10.43.138.2:/tmp/

Password:

host\_list 100% 30 83.2KB/s 00:00

If you do not specifiy the target directory while doing scp, then the home directory of the target user will be used are destination.

# scp -q -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null host\_list deepak@10.43.138.2:

Password:

IMPORTANT NOTE:  
Make sure you always include at least the colon when copying a file, since if you don't include it, you'll end up copying the file to your current working directory instead of the target.

With our previous scp examples, we've only been copying a single file. If we want to transfer or download an entire directory and its contents, we will need to use the -r option, which allows us to do a recursive copy:

# scp -r -q -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null /home/deepak/mydir deepak@10.43.138.2:

Password:

**Using NFS to share file from one server to another**

A Network File System (**NFS**) is a great method of sharing files between Linux or UNIX servers. I have written another article with [detailed steps to setup NFSv4 and NFSv3 with examples in RHEL/CentOS 7 and 8 Linux](https://www.golinuxcloud.com/configure-nfs-server-client-rhel-centos-7-8/)

On my RHEL node I have installed nfs-utils and now will setup my exports. To set up NFS, let's first create some directories that we will share to other users. Each share in NFS is known as an Export.

# cat /etc/exports

/share \*(rw,no\_root\_squash)

/share 10.0.2.0/255.255.255.0(rw,no\_root\_squash)

* In the first line I have given share access to world for /share directory.
* In the second line, after the directory is called out in a line, we're also setting which network is able to access them (10.0.2.0/255.255.255.0 in our case). This means that if you're connecting from a different network, your access will be denied.
* As far as what these options do, the first (rw) is rather self-explanatory.
* One option you'll see quite often in the wild is no\_root\_squash. Normally, the root user on one system will get map to nobody on the other for security reasons. In most cases, one system having root access to another is a bad idea. The no\_root\_squash option disables this, and it allows the root user on one end to be treated as the root user on the other. .

NOTE:  
Check the man pages for export for more information on additional options you can pass to your exports.

Next restart your nfs-server services on the server

# systemctl restart nfs-server.service

To check the list of shares currently exported

# exportfs -v

/share 10.0.2.0/255.255.255.0(rw,sync,wdelay,hide,no\_subtree\_check,sec=sys,secure,no\_root\_squash,no\_all\_squash)

/share <world>(rw,sync,wdelay,hide,no\_subtree\_check,sec=sys,secure,no\_root\_squash,no\_all\_squash)

Now try to mount the directory /share from the client side

[root@node1 ~]# mount -t nfs node2:/share /mnt

So our directory mount is successful, Next validate the content

[root@node1 ~]# cd /mnt/

[root@node1 mnt]# ls

test1 test2

We can validate the same on our client node using below command

[root@node1 mnt]# mount | grep share

node2:/share on /mnt type nfs4 (rw,relatime,vers=4.1,rsize=131072,wsize=131072,namlen=255,hard,proto=tcp,port=0,timeo=600,retrans=2,sec=sys,clientaddr=10.0.2.20,local\_lock=none,addr=10.0.2.21)

NOTE:  
For our examples we have disabled the firewall and selinux. So if you face any issues, check your iptables and selinux rules and [add iptables rules to allow ssh connection](https://www.golinuxhub.com/2014/03/how-to-allowblock-ssh-connection-from.html).

After successfully [mounting the share on the client](https://www.golinuxcloud.com/show-nfs-shares-list-nfs-client-mount-points/) node, you can copy the file locally to your node.

# cp -av /mnt/\* /tmp/

‘/mnt/test1’ -> ‘/tmp/test1’

‘/mnt/test2’ -> ‘/tmp/test2’

**Using SSHFS to copy file from one server to another**

SSH Filesystem (**SSHFS**) is a file sharing solution similar to NFS and Samba. NFS and Samba are great solutions for designating file shares but these technologies may be more complex than necessary if you want to set up a temporary file-sharing service to use for a specific period of time. SSHFS allows you to mount a remote directory on your local machine, and have it treated just like any other directory. The mounted [SSHFS directory will be available for the life of the SSH connection](https://www.golinuxcloud.com/openssh-authentication-methods-sshd-config/) and can be used to copy file from one server to another.

**Drawbacks of using SSHFS**

* Performance of file transfers won't be as fast as with an NFS mount, since there's encryption that needs to be taken into consideration as well
* Another downside is that you'd want to save your work regularly as you work on files within an SSHFS mount, because if the SSH connection drops for any reason, you may lose data.

SSHFS is part of [EPEL repository](https://www.golinuxcloud.com/install-epel-repo-in-rhel-8-centos-dnf-yum/), which you can install using yum

# yum -y install sshfs

For SSHFS to work, we'll need a directory on both your local Linux machine as well as a remote Linux server. SSHFS can [mount any directory](https://www.golinuxcloud.com/linux-mount-command-iso-usb-network-drive/) from the remote server where you have SSH access.

Here I am mounting /share from node2 on node1

[root@node1 ~]# sshfs root@node2:/share /mnt

root@node2's password:

Now validate the content of /mnt and make sure the path is properly mounted

[root@node1 ~]# cd /mnt/

[root@node1 mnt]# ls

test1 test2

[root@node1 mnt]# mount | grep share

root@node2:/share on /mnt type fuse.sshfs (rw,nosuid,nodev,relatime,user\_id=0,group\_id=0)

Now you can copy the files from /mnt locally to any other directory

[root@node1 mnt]# cp -av \* /tmp/

‘test1’ -> ‘/tmp/test1’

‘test2’ -> ‘/tmp/test2’

Once your copying is complete, manually unmount the respective directory. There are two ways to do so. First, we can use the umount command as the root (just like we normally would):

[root@node1 ~]# umount /mnt/

You can also use [**HTTPS for file sharing**](https://www.golinuxcloud.com/securely-transfer-files-https-curl-ssl-linux/), although it would be more like of file uploading and downloading via **GET** and **PUT** using curl command.

Lastly I hope the commands from this article to copy file from one server to another in Linux or Unix was helpful. So, let me know your suggestions and feedback using the comment section.