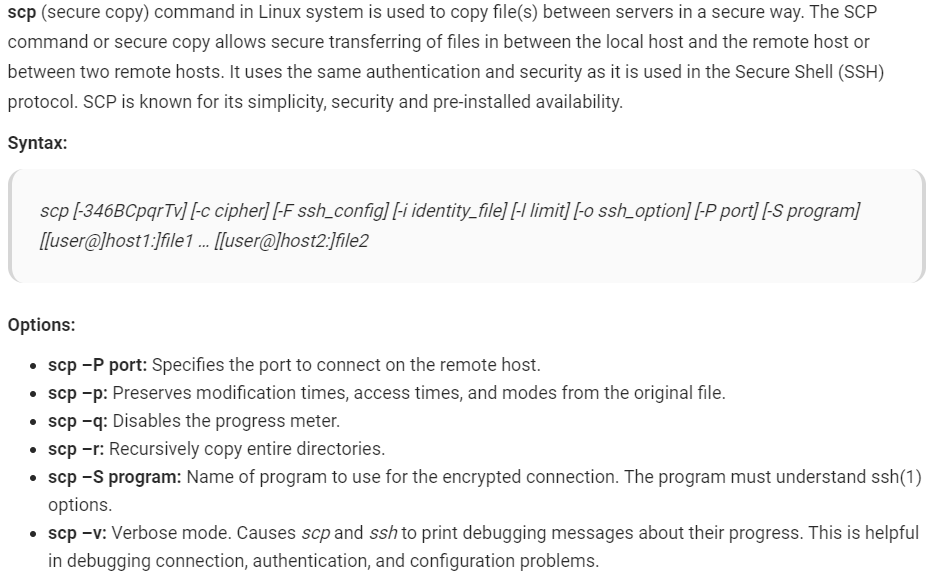
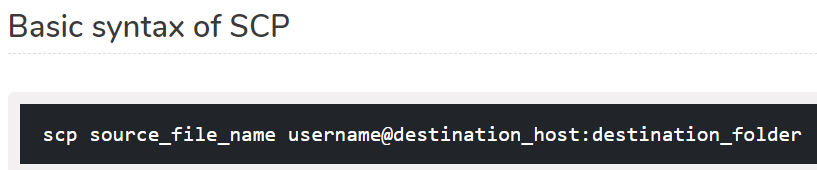
**scp command:**



* If you haven’t set a passwordless SSH login to the remote machine, you will be asked to enter the user password when we use scp command.



To copy a file from local to remote server.

* **scp file.txt** [**remote\_username@10.10.0.2:/remote/directory**](mailto:remote_username@10.10.0.2:/remote/directory)

to copy a directory from local to remote server.

* **scp -r /local/directory remote\_username@10.10.0.2:/remote/directory**

by default, ssh uses port 22. But if we want to use any other port, we need to use the below command.

* **scp -P 2322 file.txt** [**remote\_username@10.10.0.2:/remote/directory**](mailto:remote_username@10.10.0.2:/remote/directory)

below is the command to copy the file from remote server to local. It is like pulling from other servers to current server.

* **scp remote\_username@10.10.0.2:/remote/file.txt /local/directory**

below is the command to copy a directory from remote server to local.

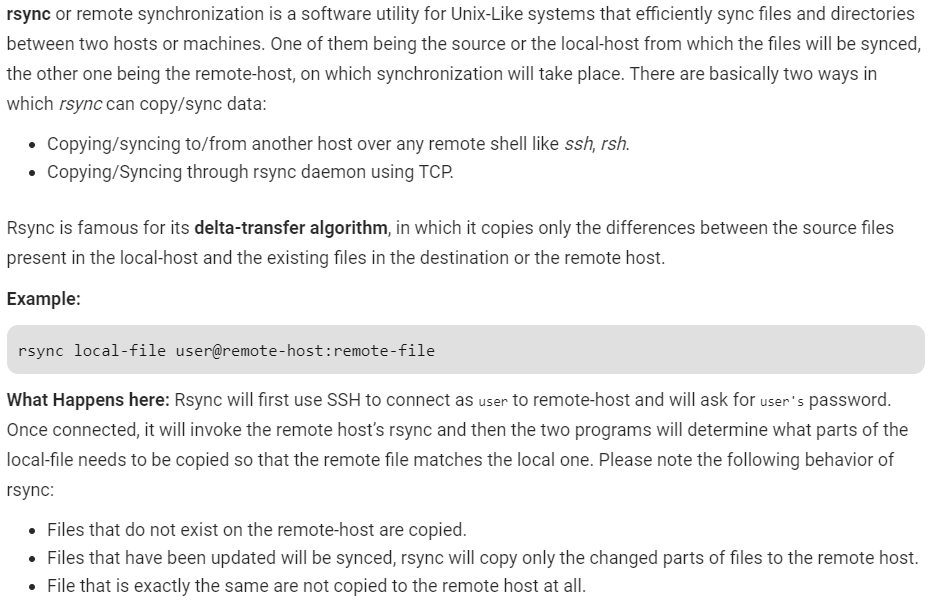
* **scp -r remote\_username@hostname:<directory path> <local path>**

we can also copy the files between two servers without even logging into as below.

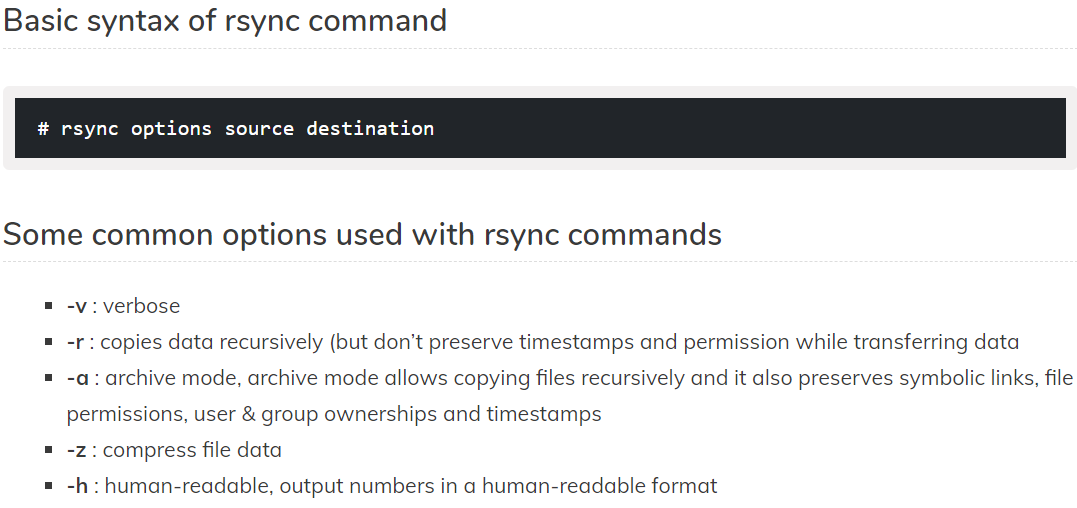
You will be prompted to enter the passwords for both remote accounts. The data will be transfer directly from one remote host to the other.

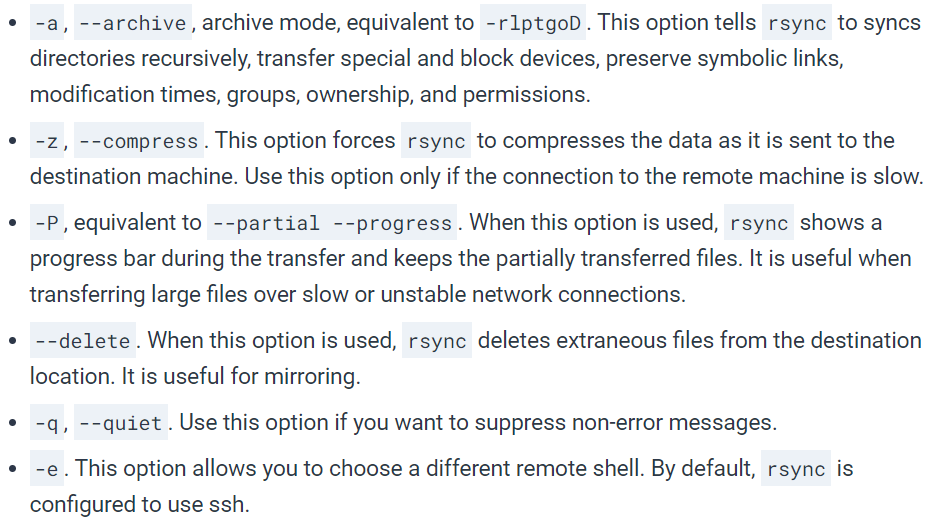
* **scp user1@host1.com:/files/file.txt** [**user2@host2.com:/files**](mailto:user2@host2.com:/files)

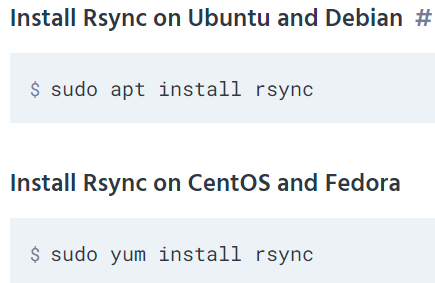
**rsync command:**



* It’s faster than scp (Secure Copy) because rsync uses remote-update protocol which allows to transfer just the differences between two sets of files. First time, it copies the whole content of a file or a directory from source to destination but from next time, it copies only the changed blocks and bytes to the destination.

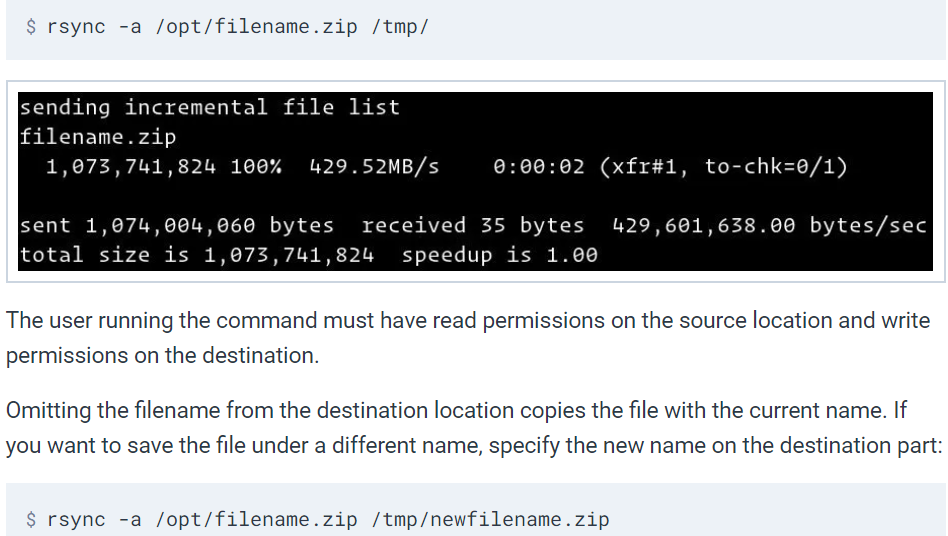






We can use rsync command to copy from local to local, local to remote & also from remote to local.

**Examples:**



If the destination directory doesn’t exist, rsync will create it.

To use rsync, it must be installed in remote server as well.

If we haven’t enabled passwordless authentication, rsync will ask for the password at runtime.

By default, it connects to remote host with port 22. If SSH on the remote server is listening to different port, then we need to use the below command.

* **rsync -a -e "ssh -p 2322" /opt/media/ remote\_user@remote\_host\_or\_ip:/opt/media/**

When transferring large amounts of data it is recommended to run the rsync command inside a screen session or to use the -P option:

* **rsync -a -P remote\_user@remote\_host\_or\_ip:/opt/media/ /opt/media/**

below is the command to exclude files and directories.

* **rsync -a --exclude=node\_modules --exclude=tmp /src\_directory/ /dst\_directory/**

we can use the below command if we want to exclude the data from a file.

* **rsync -a --exclude-from='/exclude-file.txt' /src\_directory/ /dst\_directory/**

we can also use include options as below.

* **rsync -avze ssh --include 'R\*' --exclude '\*' root@192.168.0.101:/var/lib/rpm/ /root/rpm**

setting the max size to be transferred in rsync.

* **rsync -avzhe ssh --max-size='200k' /var/lib/rpm/** [**root@192.168.0.100:/root/tmprpm**](mailto:root@192.168.0.100:/root/tmprpm)

below is the command to delete source files automatically.

* **rsync --remove-source-files -zvh backup.tar /tmp/backups/**

all these above options can also be used to copy from remote to local with below syntax:

* **rsync <remote host>:<path> <local path>**

we can use the below command to ignore the files if already exists in destination.

* **rsync -a --ignore-existing /local/directory/ host:/remote/directory/**
* **rsync -u <file> <remote server>:<path>**