Syncing Files with Rsync

The `rsync` command in linux allows us to sync files both locally as well as remotely across different directories and remote machines.

Why `rsync`?

The important thing to understand here is that `rsync` only copies (synchronises) what's different (somewhat like git). This means that every time something is updated in a directory at a particular location, only the *changed* portions since the last backup will get synced, instead of again copying the entire directory.

Another scenario would be when you lose your network connection in the middle of taking a backup. Even in this case `rsync` will know where it left off, and continue from there.

Lets Begin

We start our exploration of `rsync` by first creating 2 directories on our local machine called `original` and `backup`.

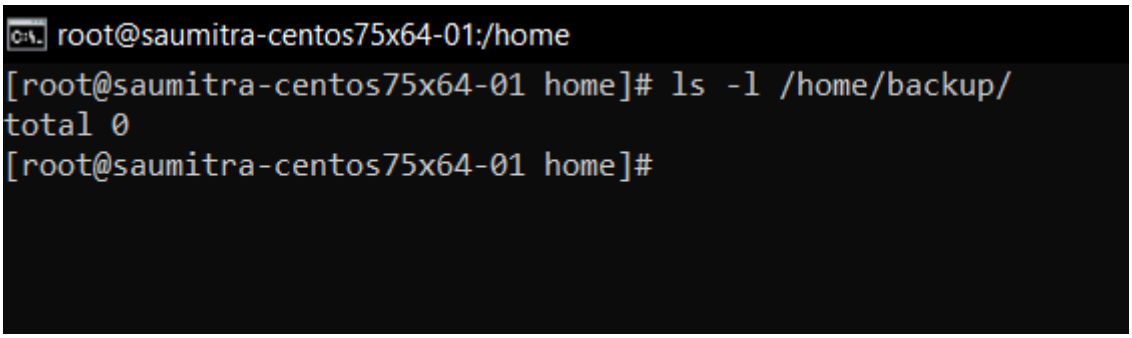
We add some important files to our `original` directory.

```
root@saumitra-centos75x64-01:/home
[root@saumitra-centos75x64-01 original]# cd ..
[root@saumitra-centos75x64-01 home]# ls -l /home/original/
total 8
-rw-r--r--. 1 root root 67 Jul 14 08:09 imp_file_1.txt
-rw-r--r--. 1 root root 68 Jul 14 08:10 imp_file_2.txt
[root@saumitra-centos75x64-01 home]#
```

These are the files that we wish to keep a backup of. Let's see what these files contain -

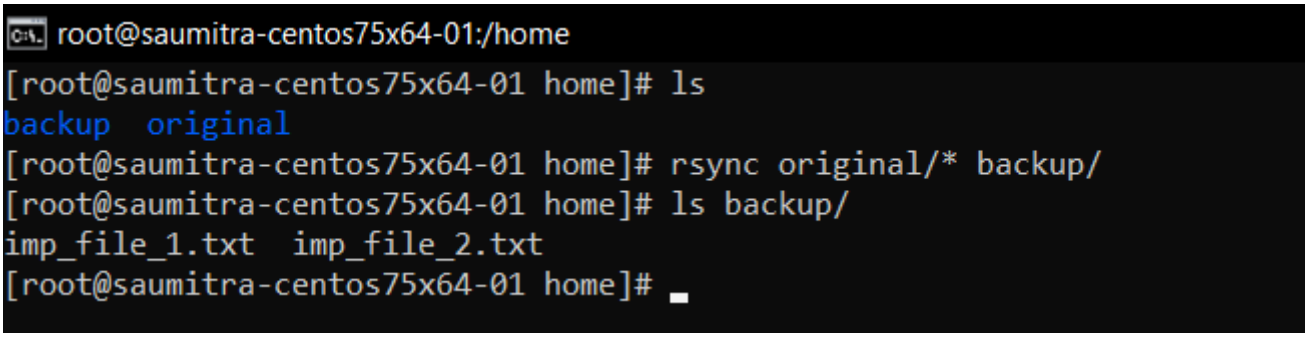
```
root@saumitra-centos75x64-01:/home
[root@saumitra-centos75x64-01 home]# cat /home/original/imp_file_1.txt
This is a very important file...
I'll be mad if I lose this... >:(
[root@saumitra-centos75x64-01 home]# cat /home/original/imp_file_2.txt
I guess I have a lot of important stuff..

DONT LOSE THIS FILE!! :O
[root@saumitra-centos75x64-01 home]#
```

Our backup folder is still empty. We can check it using the `ls` command.

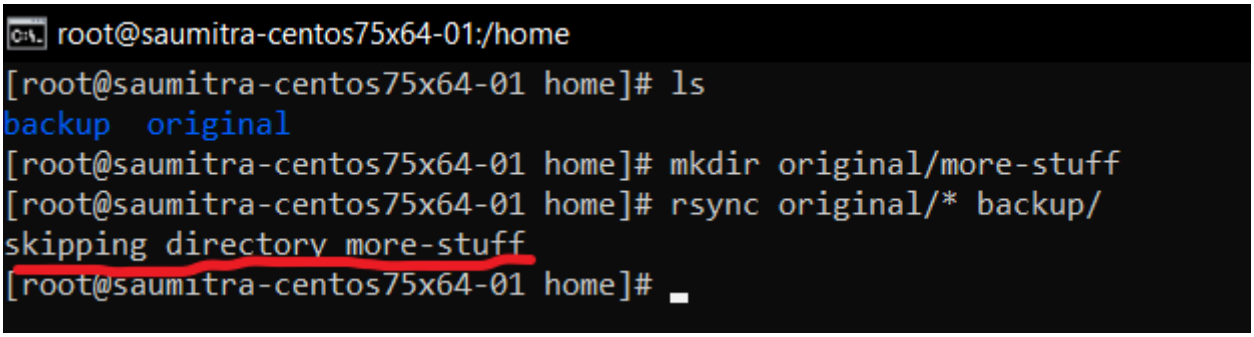


In order to backup the contents of the `original` directory to the `backup` folder, we simply run the `rsync` command, specifying the source and destination directories.

```
rsync source_path dest_path
```

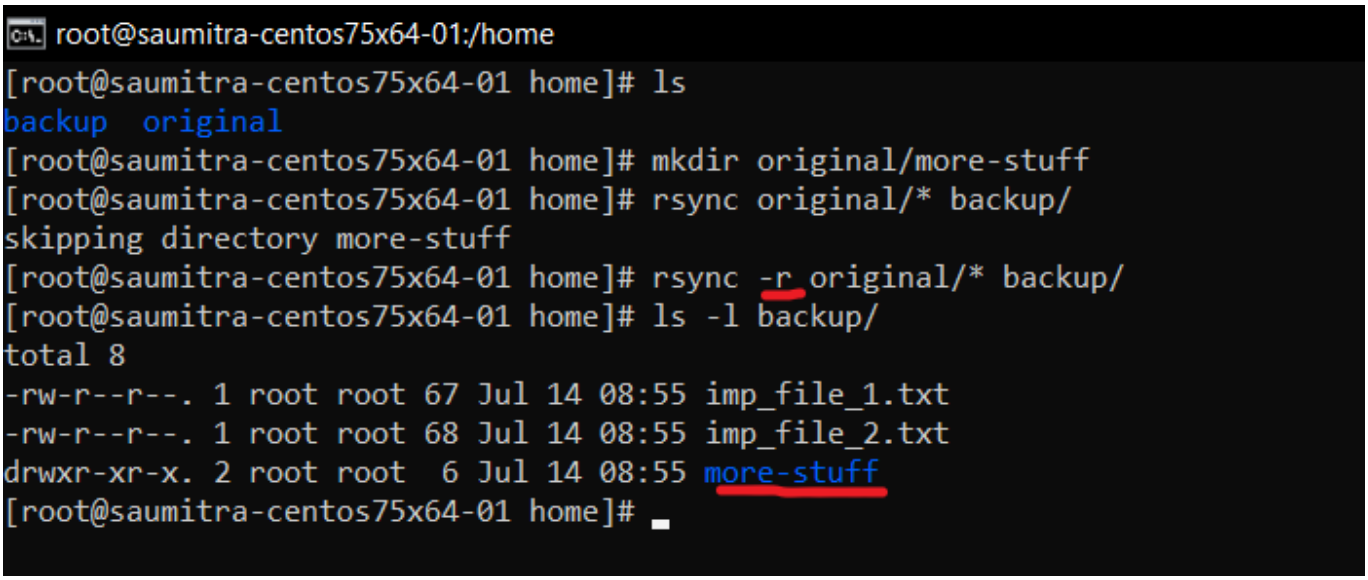


Its done! We can check our backup directory to see that the files have been copied! Notice that we had to add a star `*` while specifying the source directory. This is because, by default `rsync` does not copy files *recursively*. This means that now if we add a new folder to `original`, `rsync` will simply ignore it.



Recursive backups (`-r` and `-a`)

In order to explicitly tell `rsync` that we want to backup all files and folders in the source directory in a *recursive* manner, we use the `-r` (recursive) option.



An analogous option is `-a` (archive). This also copies all the SYMLINKS and timestamps, and is generally the more preferred option.

Dry Run (`--dry-run`)

Consider a scenario when you are ready to back up gigabytes of data. But after performing the backup, you realise that this was not the destination you wanted to back up to!

Well, you just wasted a lot of time, bandwidth and effort in doing the wrong thing!

As a precautionary measure, we can verify what files we want to copy to the destination using the `dry-run` option. This makes rsync output the same exact things that it would have done while copying the files, but it will not actually copy those files...

```
root@saumitra-centos75x64-01:/home
[root@saumitra-centos75x64-01 home]# ls -l ./original/
total 8
-rw-r--r--. 1 root root  0 Jul 14 09:04 imp_file_10.txt
-rw-r--r--. 1 root root 67 Jul 14 08:09 imp_file_1.txt
-rw-r--r--. 1 root root 68 Jul 14 08:10 imp_file_2.txt
-rw-r--r--. 1 root root  0 Jul 14 09:03 imp_file_3.txt
-rw-r--r--. 1 root root  0 Jul 14 09:03 imp_file_4.txt
-rw-r--r--. 1 root root  0 Jul 14 09:03 imp_file_5.txt
-rw-r--r--. 1 root root  0 Jul 14 09:03 imp_file_6.txt
-rw-r--r--. 1 root root  0 Jul 14 09:03 imp_file_7.txt
-rw-r--r--. 1 root root  0 Jul 14 09:04 imp_file_8.txt
-rw-r--r--. 1 root root  0 Jul 14 09:04 imp_file_9.txt
drwxr-xr-x. 2 root root  6 Jul 14 08:54 more-stuff
[root@saumitra-centos75x64-01 home]#
[root@saumitra-centos75x64-01 home]#
[root@saumitra-centos75x64-01 home]#
[root@saumitra-centos75x64-01 home]# rsync -a -v --dry-run original/ backup/
sending incremental file list
./
imp_file_10.txt
imp_file_3.txt
imp_file_4.txt
imp_file_5.txt
imp_file_6.txt
imp_file_7.txt
imp_file_8.txt
imp_file_9.txt

sent 344 bytes  received 44 bytes  776.00 bytes/sec
total size is 135  speedup is 0.35 (DRY RUN)
[root@saumitra-centos75x64-01 home]#
```

So, we get a small glimpse into the future, before we actually execute the backup process. To backup files, we simply remove the `--dry-run` option, and the command works normally.

```
[root@saumitra-centos75x64-01 home]# rsync -a -v original/ backup/
sending incremental file list
./
imp_file_10.txt
imp_file_3.txt
imp_file_4.txt
imp_file_5.txt
imp_file_6.txt
imp_file_7.txt
imp_file_8.txt
imp_file_9.txt

sent 632 bytes  received 172 bytes  1,608.00 bytes/sec
total size is 135  speedup is 0.17
[root@saumitra-centos75x64-01 home]#
```

Only Differences

As stated earlier, `rsync` only copies the files that are changed in the original directory. It will not re-copy everything. So, if we delete some files in the `original` directory, `rsync` has the ability to detect that.

```
[root@saumitra-centos75x64-01 home]#
[root@saumitra-centos75x64-01 home]# rm -rf ./backup/imp_file_10.txt
[root@saumitra-centos75x64-01 home]# rm -rf ./backup/imp_file_9.txt
[root@saumitra-centos75x64-01 home]# rm -rf ./backup/imp_file_8.txt
[root@saumitra-centos75x64-01 home]#
[root@saumitra-centos75x64-01 home]#
[root@saumitra-centos75x64-01 home]# rsync -a -v --dry-run original/ backup/
sending incremental file list
./
imp_file_10.txt
imp_file_8.txt
imp_file_9.txt

sent 329 bytes  received 29 bytes  238.67 bytes/sec
total size is 135  speedup is 0.38 (DRY RUN)
[root@saumitra-centos75x64-01 home]#
```

However, the reverse is not true. If we add a file to the `backup` directory, rsync will not add it to the `original` directory.

```
root@saumitra-centos75x64-01:/home
[root@saumitra-centos75x64-01 home]# touch ./backup/test.txt
[root@saumitra-centos75x64-01 home]# rsync -a -v --dry-run original/ backup/
sending incremental file list
./

sent 320 bytes  received 20 bytes  680.00 bytes/sec
total size is 135  speedup is 0.40 (DRY RUN)
[root@saumitra-centos75x64-01 home]#
```

Delete (`-delete`)

If we want our `backup` directory to mirror the `original` directory, we can add the `-delete` option during `rsync`. This option will delete all the files in the `backup` directory that are not there in the `original` directory.

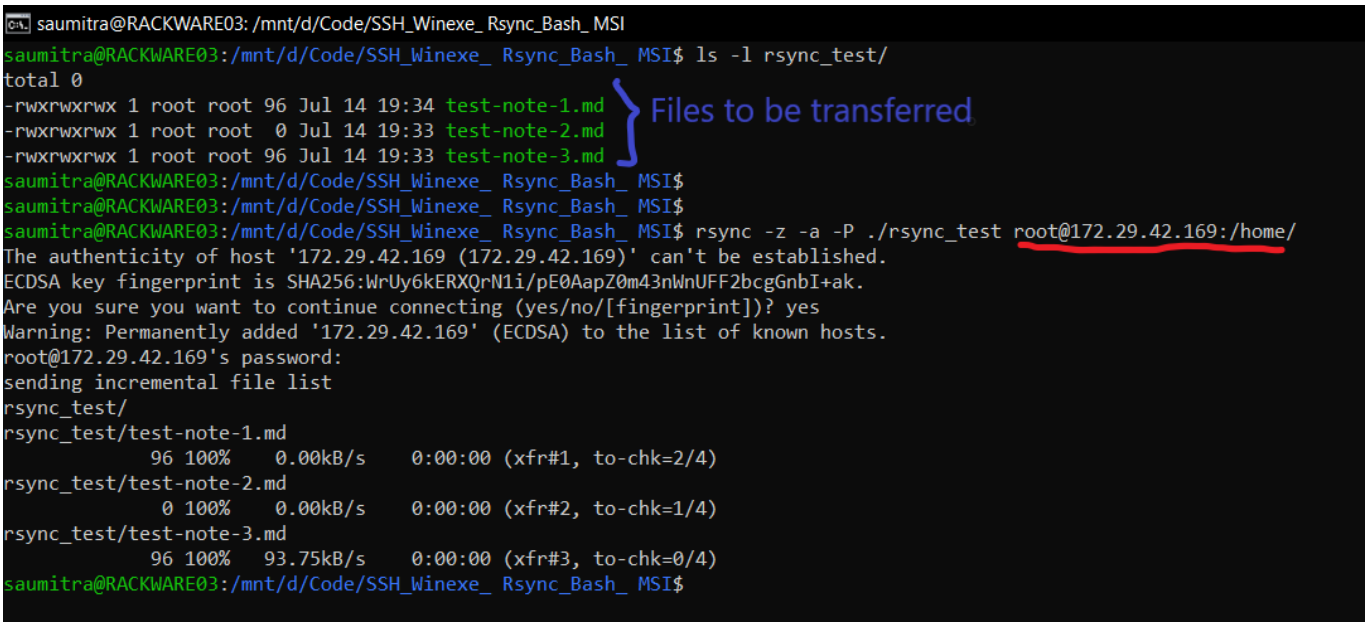
```
[root@saumitra-centos75x64-01 home]# rsync -a -v --delete --dry-run original/ backup/
sending incremental file list
deleting test.txt
./

sent 320 bytes  received 32 bytes  704.00 bytes/sec
total size is 135  speedup is 0.38 (DRY RUN)
[root@saumitra-centos75x64-01 home]#
```

```
CAUTION-
The delete option comes with a very dangerous weapon. If we ever make a mistake of executing the rsync command with the -delete option with an empty source directory, it (as expected) clear the contents of the backup directory too. Many a tale have been told of titams falling to their knees regretting using the -delete option with rsync. This is the reason we must always --dry-run before execution.
```

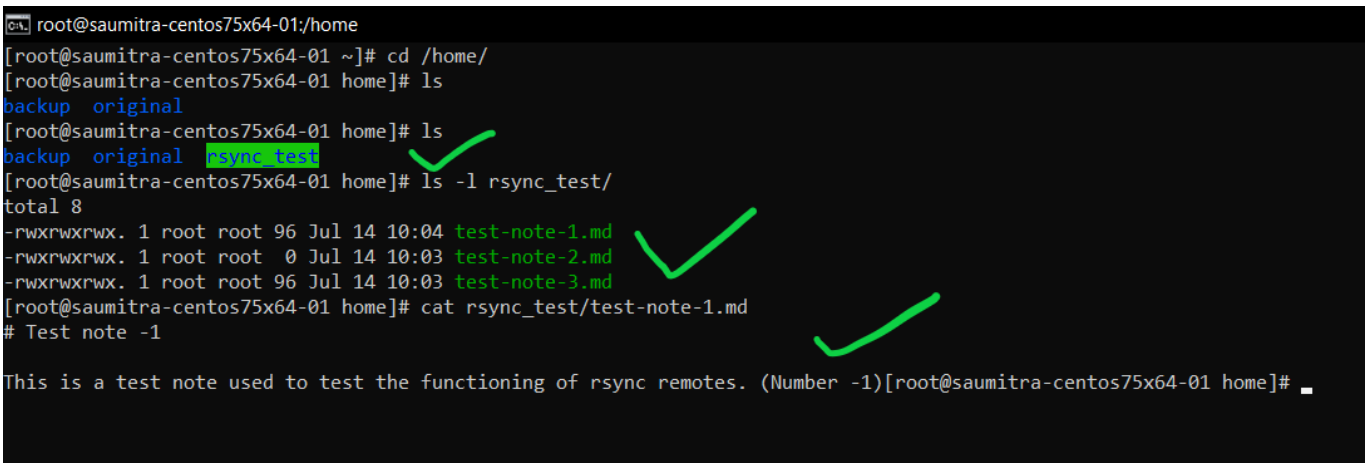
Remote File Transfer

`rsync` as a tool really shines here. It gives us the ability to back-up data between remote machines. The syntax of doing so is very similar to that of `ssh`. The `-z` option allows us to compress files before sending. The `-P` option shows the transfer progress during the operation.

```
rsync -z -a -P source_path user@ip_address:dest_path
```



We can check that we got the files on our remote machine -



References -

1. https://www.youtube.com/watch?v=qE77MbDnljA
2. https://www.youtube.com/watch?v=2PnAohLS-Q4
3. https://stackoverflow.com/questions/818255/in-the-shell-what-does-21-mean
4. https://serverfault.com/questions/835104/what-does-connecttimeout-in-ssh-config-mean
5. https://www.perturb.org/display/1208_SSH_Batch_mode_for_SSH_SCP_scripting.html
6. https://stackoverflow.com/questions/7411052/bash-while-loop-iterates-only-once-whenever-body-contains-ssh