

Rsync

Rsync (Remote Sync) is a most commonly used command for copying and synchronizing files and directories remotely as well as locally in Linux/Unix systems. With the help of rsync command you can copy and synchronize your data remotely and locally across directories, across disks and networks, perform data backups and mirroring between two Linux machines.

Advantages and features of Rsync command:

It efficiently copies and sync files to or from a remote system.

- Supports copying links, devices, owners, groups and permissions.
- 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.
- Rsync consumes less bandwidth as it uses compression and decompression method while sending and receiving data both ends.

Basic syntax of rsync command

rsync options source destination

Some common options used with rsync commands

-v : verbose

-r : copies data recursively (but don't preserve timestamps and permission while transferring data)

-a : archive mode, archive mode allows copying files recursively and it also preserves symbolic links, file permissions, user & group ownerships and timestamps

-z : compress file data

-h : human-readable, output numbers in a human-readable format

Install rsync in your Linux machine

We can install rsync package with the help of following command.

yum install rsync (On Red Hat based systems)

1.Copy/Sync Files and Directory Locally

This following command will sync a single file on a local machine from one location to another location. Here in this example, a file name backup.tar needs to be copied or synced to /tmp/backups/ folder.

```
[root@tecmint]# rsync -zvh backup.tar /tmp/backups/  
created directory /tmp/backups/  
backup.tar  
sent 14.71M bytes received 31 bytes 3.27M bytes/sec  
total size is 16.18M speedup is 1.10
```

In above example, you can see that if the destination is not already exists rsync will create a directory automatically for destination.

Copy/Sync a Directory on Local Computer

The following command will transfer or sync all the files of from one directory to a different directory in the same machine. Here in this example, /root/rmpkgs contains some rpm package files and you want that directory to be copied inside /tmp/backups/ folder.

```
[root@tecmint]# rsync -avzh /root/rmpkgs /tmp/backups/
sending incremental file list
rmpkgs/
rmpkgs/httpd-2.2.3-82.el5.centos.i386.rpm
rmpkgs/mod_ssl-2.2.3-82.el5.centos.i386.rpm
rmpkgs/nagios-3.5.0.tar.gz
rmpkgs/nagios-plugins-1.4.16.tar.gz
sent 4.99M bytes  received 92 bytes  3.33M bytes/sec
total size is 4.99M  speedup is 1.00
```

2. Copy/Sync Files and Directory to or From a Server

a) Copy a Directory from Local Server to a Remote Server

This command will sync a directory from a local machine to a remote machine. For example: There is a folder in your local computer “rmpkgs” which contains some RPM packages and you want that local directory’s content send to a remote server, you can use following command.

```
[root@tecmint]$ rsync -avz rmpkgs/ root@192.168.0.101:/home/
root@192.168.0.101's password:
sending incremental file list
./
httpd-2.2.3-82.el5.centos.i386.rpm
mod_ssl-2.2.3-82.el5.centos.i386.rpm
nagios-3.5.0.tar.gz
nagios-plugins-1.4.16.tar.gz
sent 4993369 bytes  received 91 bytes  399476.80 bytes/sec
total size is 4991313  speedup is 1.00
```

b) Copy/Sync a Remote Directory to a Local Machine

This command will help you sync a remote directory to a local directory. Here in this example, a directory /home/tarunika/rmpkgs which is on a remote server is being copied in your local computer in /tmp/myrpms.

```
[root@tecmint]# rsync -avzh root@192.168.0.100:/home/tarunika/rmpkgs /tmp/myrpms
```

```
root@192.168.0.100's password:
receiving incremental file list
created directory /tmp/myrpms
rpmpkgs/
rpmpkgs/httpd-2.2.3-82.el5.centos.i386.rpm
rpmpkgs/mod_ssl-2.2.3-82.el5.centos.i386.rpm
rpmpkgs/nagios-3.5.0.tar.gz
rpmpkgs/nagios-plugins-1.4.16.tar.gz
sent 91 bytes received 4.99M bytes 322.16K bytes/sec
total size is 4.99M speedup is 1.00
```

3. Rsync Over SSH

With rsync, we can use SSH (Secure Shell) for data transfer, using SSH protocol while transferring our data you can be ensured that your data is being transferred in a secured connection with encryption so that nobody can read your data while it is being transferred over the wire on the internet.

Also when we use rsync we need to provide the user/root password to accomplish that particular task, so using SSH option will send your logins in an encrypted manner so that your password will be safe.

a) Copy a File from a Remote Server to a Local Server with SSH

To specify a protocol with rsync you need to give “-e” option with protocol name you want to use. Here in this example, We will be using “ssh” with “-e” option and perform data transfer.

```
[root@tecmint]# rsync -avzhe ssh root@192.168.0.100:/root/install.log /tmp/
root@192.168.0.100's password:
receiving incremental file list
install.log
sent 30 bytes received 8.12K bytes 1.48K bytes/sec
total size is 30.74K speedup is 3.77
```

b) Copy a File from a Local Server to a Remote Server with SSH

```
[root@tecmint]# rsync -avzhe ssh backup.tar root@192.168.0.100:/backups/
root@192.168.0.100's password:
sending incremental file list
backup.tar
sent 14.71M bytes received 31 bytes 1.28M bytes/sec
total size is 16.18M speedup is 1.10
```

4. Show Progress While Transferring Data with rsync

To show the progress while transferring the data from one machine to a different machine, we can use ‘-progress’ option for it. It displays the files and the time remaining to complete the transfer.

```
[root@tecmint]# rsync -avzhe ssh --progress /home/rpmpkgs root@192.168.0.100:/root/rpmpkgs
root@192.168.0.100's password:
sending incremental file list
created directory /root/rpmpkgs
rpmpkgs/
```

```
rpm_pkgs/httpd-2.2.3-82.el5.centos.i386.rpm
1.02M 100%    2.72MB/s    0:00:00 (xfer#1, to-check=3/5)
rpm_pkgs/mod_ssl-2.2.3-82.el5.centos.i386.rpm
99.04K 100%  241.19kB/s    0:00:00 (xfer#2, to-check=2/5)
rpm_pkgs/nagios-3.5.0.tar.gz
1.79M 100%    1.56MB/s    0:00:01 (xfer#3, to-check=1/5)
rpm_pkgs/nagios-plugins-1.4.16.tar.gz
2.09M 100%    1.47MB/s    0:00:01 (xfer#4, to-check=0/5)
sent 4.99M bytes received 92 bytes 475.56K bytes/sec
total size is 4.99M speedup is 1.00
```

5. Use of `--include` and `--exclude` Options

These two options allow us to include and exclude files by specifying parameters with these options. They help us to specify those files or directories which you want to include in your sync and exclude files and folders which you don't want to be transferred.

Here in this example, the rsync command will include those files and directory only which starts with 'R' and exclude all other files and directory.

```
[root@tecmint]# rsync -avze ssh --include 'R*' --exclude '*' root@192.168.0.101:/var/lib/rpm/
/root/rpm
root@192.168.0.101's password:
receiving incremental file list
created directory /root/rpm
./
Requirename
Requireversion
sent 67 bytes received 167289 bytes 7438.04 bytes/sec
total size is 434176 speedup is 2.59
```

6. Use of `--delete` Option

If a file or directory does not exist at the source, but already exists at the destination, you might want to delete that existing file/directory at the target while syncing.

We can use the `--delete` option to delete files that are not there in source directory.

Source and target are in sync. Now creating new file test.txt at the target.

```
[root@tecmint]# touch test.txt
[root@tecmint]# rsync -avz --delete root@192.168.0.100:/var/lib/rpm/ .
Password:
receiving file list ... done
deleting test.txt
./
sent 26 bytes received 390 bytes 48.94 bytes/sec
total size is 45305958 speedup is 108908.55
Target has the new file called test.txt, when synchronize with the source with '--delete' option, it
removed the file test.txt.
```

7. Set the Max Size of Files to be transferred

You can specify the Max file size to be transferred or sync. You can do it with “--max-size” option. Here in this example, Max file size is 200k, so this command will transfer only those files which are equal or smaller than 200k.

```
[root@tecmint]# rsync -avzhe ssh --max-size='200k' /var/lib/rpm/ root@192.168.0.100:/root/tmprpm
root@192.168.0.100's password:
sending incremental file list
created directory /root/tmprpm
./
Conflictname
Group
Installtid
Name
Provideversion
Pubkeys
Requireversion
Sha1header
Sigmd5
Triggername
__db.001
sent 189.79K bytes received 224 bytes 13.10K bytes/sec
total size is 38.08M speedup is 200.43
```

8. Automatically Delete source Files after successful Transfer

Now, suppose you have a main web server and a data backup server, you created a daily backup and synced it with your backup server, now you don't want to keep that local copy of backup in your web server. So, will you wait for transfer to complete and then delete those local backup file manually? Of Course NO. This automatic deletion can be done using ‘--remove-source-files’ option.

```
[root@tecmint]# rsync --remove-source-files -zvh backup.tar /tmp/backups/
backup.tar
sent 14.71M bytes received 31 bytes 4.20M bytes/sec
total size is 16.18M speedup is 1.10
[root@tecmint]# ll backup.tar
ls: backup.tar: No such file or directory
```

9. Do a Dry Run with rsync

If you are a newbie and using rsync and don't know what exactly your command going do. Rsync could really mess up the things in your destination folder and then doing an undo can be a tedious job.

Use of this option will not make any changes only do a dry run of the command and shows the output of the command, if the output shows exactly same you want to do then you can remove ‘--dry-run’ option from your command and run on the terminal.

```
root@tecmint]# rsync --dry-run --remove-source-files -zvh backup.tar /tmp/backups/
backup.tar
sent 35 bytes received 15 bytes 100.00 bytes/sec
total size is 16.18M speedup is 323584.00 (DRY RUN)
```

10. Set Bandwidth Limit and Transfer File

You can set the bandwidth limit while transferring data from one machine to another machine with the help of ‘--bwlimit’ option. This options helps us to limit I/O bandwidth.

```
[root@tecmint]# rsync --bwlimit=100 -avzhe ssh /var/lib/rpm/ root@192.168.0.100:/root/tmprpm/  
root@192.168.0.100's password:  
sending incremental file list  
sent 324 bytes received 12 bytes 61.09 bytes/sec  
total size is 38.08M speedup is 113347.05
```

Also, by default rsync syncs changed blocks and bytes only, if you want explicitly want to sync whole file then you use ‘-W’ option with it.

```
[root@tecmint]# rsync -zvhw backup.tar /tmp/backups/backup.tar  
backup.tar  
sent 14.71M bytes received 31 bytes 3.27M bytes/sec  
total size is 16.18M speedup is 1.10
```