

Sponsored by



UpCloud

Click here

to get \$25 and Cloud Server

# Backup and Restore Your Linux System with rsync

```
[alu@archPC ~]$ sudo rsync -aXv --delete --exclude=/dev/* --exclude=/proc/* --exclude=/sys/* --exclude=/tmp/* --exclude=/run/* --exclude=/mnt/* --exclude=/media/* --exclude="swapfile" --exclude="lost+found" --exclude=".cache" --exclude="Downloads" --exclude=".VirtualBoxVMs" --exclude=".ecryptfs" / /run/media/alu/ALU/
```

Create a backup with rsync

If you need to backup your system, there is no better way than making a backup with rsync.

Rsync (Remote Sync) is a popular and powerful tool used to copy and synchronize files and directories between remote or local Linux/Unix systems. With the help of rsync, we can easily copy/synchronize data between local and remote directories, across different drives and networks.

In this tutorial, I will show you how to make a backup with rsync using the Linux terminal. If you prefer graphical programs, [you can use rsync with graphical interface too.](#)

## Video Tutorial

Backup and Restore Your Linux System with rsync



## Making the backup with rsync

To make this tutorial, I will use Arch Linux in a virtual machine built with VirtualBox. To simulate an external hard drive, I will connect a USB flash drive where the backup will be stored and then restored. I recommend you do the same to test your backup. This will give you the confidence to know that your backup works because **an untested backup is not a backup.**

For this case we will use this entire command:

```
sudo rsync -aAXv --delete --dry-run --exclude=/dev/* --exclude=/proc/* --
exclude=/sys/* --exclude=/tmp/* --exclude=/run/* --exclude=/mnt/* --
exclude=/media/* --exclude="swapfile" --exclude="lost+found" --exclude=".cache"
--exclude="Downloads" --exclude=".VirtualBoxVMs"--exclude=".ecryptfs" /
/run/media/alu/ALU/
```

To make a backup with rsync, we usually use the command line. I know that not everyone is confident with the command line tools, but you will realize that the process is not that complicated and you can also back up your system using the command line.

We now proceed to explain what this command means:

`sudo` - to execute the command as a superuser. Mandatory use.

`rsync` - is the program itself to use.

`-a` - archive mode.

`-A` - preserve Access Control List.

`-X` - preserve extended attributes.

Basically, these three options mean to preserve all the attributes of your files. Owner attributes or permissions will not be modified during the backup process.

`-v` - It will show the progress of the backup.

`--delete` - this option allows you to make an incremental backup. That means, if it is not your first backup, it will backup only the difference between your source and the destination. So, it will backup only new files and modified files and it will also delete all the files in the backup which were deleted on your system. Be careful with this option.

`--dry-run` - This option simulates the backup. Useful to test its execution.

`--exclude` - Excludes folders and files from backup. I typed exclude as a separate option for every directory. You can also use it this way `--exclude={/dev/*,/proc/*,/sys/*,/tmp/*,/run/*,/mnt/*,/media/*,/home/*,/lost+found}`. But make sure you change your working directory to root (`cd /`) before you run rsync, otherwise the joint exclude option may not work.

The excluded folders depend directly on each of us, however, the `/dev/`, `/proc/`, `/proc/` `/sys/` `/tmp/` `/run/` `/mnt/` and `/media` folders are not important to backup because rsync will not copy their content. `/mnt/` it is vital to exclude them if we connect a USB memory.

`/` - What we want to back up.

`/run/media/alu/ALU` - This is where you want to backup. I recommend to [encrypt the destination](#), to make your data safe.

We press enter, the command will execute in a simulation mode (because of the `--dry-run` option) This way we test it to make sure everything is okay. When you're sure that everything is performed as you want, you remove `--dry-run` from the command and run it again.

Note: It is recommended that the backup drive has a Linux compatible file system as *ext4*.

## Restore the backup with rsync

To restore the backup we have made, we are going to boot from a live ISO. Since we are working with Arch Linux, then the iso image must be from Arch Linux. Next, we must mount our USB flash drive.

Once logged in from the live image, we must create two folders, one for the system on the hard disk and the other where the backup created will be mounted:

```
mkdir /mnt/system /mnt/usb
```

Next, we need to check the names of our devices:

```
lsblk
```

Then, we must mount the file system and the backup on the USB flash device:

```
mount /dev/sda1 /mnt/system
mount /dev/sdb1 /mnt/usb
```

```
Arch Linux 4.11.7-1-ARCH (tty1)
archiso login: root (automatic login)
root@archiso ~ # mkdir /mnt/system /mnt/usb
root@archiso ~ # lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0       7:0    0 398.3M  1 loop /run/archiso/sfs/airootfs
sda         8:0    0   10G  0 disk
└─sda1      8:1    0   10G  0 part
sdb         8:16   1   15G  0 disk
└─sdb1      8:17   1   15G  0 part
sr0         11:0    1   510M  0 rom  /run/archiso/bootmnt
root@archiso ~ # mount /dev/sda1 /mnt/system
root@archiso ~ # mount /dev/sdb1 /mnt/usb
```

Mounting directories to restore the backup with rsync

Finally, we proceed to perform the restoration of our backup. For this we run:

```
rsync -aAXv --delete --exclude="lost+found" /mnt/usb/ /mnt/system/
```

By running this command, we restore the backup of our system.

## Conclusion

Rsync is a powerful tool that can be used from the command line, with multiple options that adapt to any need. You can of course use [backup programs with graphical interface](#). There are also some really good [non-open sources graphical backup programs](#).

To make your backup safe, [I also recommend encrypting your backup drive](#).

Have you already done your backup? Let me know about your experience with rsync.

Share



Please share this article through your social networks.

04 Sep 2018



[Average Linux User](#) [Follow](#)

I am the founder of the Average Linux User project, which is a hobby I work on at night. During the day I am a scientist who uses computers to analyze genetic data.

[Linux applications](#)

[#Arch Linux](#)

[#backup](#)

[#command line](#)

[#rsync](#)

[« A step by step Arch Linux installation guide](#)

[10 Things to do first after installing Arch Linux »](#)

## Comments (12)



Show: oldest first ▼

Enter your data or log in to comment  

Your name

Your email (optional)

Write a public comment (use Ctrl + Enter to send; Markdown is supported)

← Send

Preview



**Chris Spatgen**

6 months ago

Thank you for the great article. I had to use the Restore the backup with rsync procedure today and it saved a lot of time. Thank you again!!

[Reply](#) [React](#) ...



**Average Linux User**

5 months ago  Chris Spatgen

Hi Chris, Thanks for the feedback and I am happy to hear the restore procedure worked!

[Reply](#) [React](#) ...



**Benjie**

5 months ago

Good day, May I ask 2 question, please. \* I have a SERVER1 with 300gb used and 200gb free space total of 500gb of HDD, I want to back it up to SERVER2. Does SERVER2 needs to have more than 300gb? \* If I run rsync in SERVER1 it will create a 300gb backup to the remaining 200gb free space, is that posible? Sorry for my bad english but I hope you can help me. Thank you.

[Reply](#) [React](#) ...



**Average Linux User**

5 months ago ↩ Benjie

\* SERVER2 needs 300G or more. \* SERVER1 won't be able to backup to itself because there is no enough space. You can try to use compression to fit 300G of raw data to 200G compressed but it depends on what data you have. Nevertheless, I do not recommend to keep backup and original data on the same hard-drive.

[Reply](#) [React](#) ...



**Benjie**

5 months ago ↩ Average Linux User

Thank you very much sir..

[Reply](#) [React](#) ...



**Average Linux User**

5 months ago ↩ Benjie

Thank you for reading!

[Reply](#) [React](#) ...



**Manu**

4 months ago

How does this work with a full system backup? Let's say I want to replace the whole data on one disk with a full system backup and launch from it. Are there any more steps needed? Thanks in advance

[Reply](#) [React](#) ...



**Average Linux User**

4 months ago ↩ Manu

It should work as described. You only need to install GRUB on that new disk from a Live USB.

[Reply](#) [React](#) ...



**binbashexe**

4 months ago

hello, thank you for great tut i want to ask you something ; i need to back up just system mean like fresh install without other config thing do i need to remove /mnt/ folder because i have prtition on it with big files i do not to be back up fdisk -l /dev/sda1 2048 7999487 7997440 3.8G 82 Linux swap / Solaris /dev/sda2 \* 7999488 105656319 97656832 46.6G 83 Linux >> root /dev/sda3 105656320 300967935 195311616 93.1G 83 Linux >> home /dev/sda4 300967936 625141759 324173824 154.6G 83 Linux >> /mnt/

[Reply](#) [React](#) ...



**Average Linux User**

4 months ago ↩ binbashexe

You need to exclude /mnt only if the drive where your backup is mounted to /mnt.

[Reply](#) [React](#) ...



davidA

1 week ago

Thanks for the explanation. I'm wondering why do you restore to "/mnt/system/"? I guess, in this context, it's important to understand the difference between hard link and soft link, right? In your example, the backup will affect sda1 (which is mounted to /mnt/system/ in the USB-Live version but to root ("/") on my installed system (which is about to be restored). So you just mount to "/mnt/system/" in order to address sda1, but it will still sit on ("/") in my system, right? The whole mounting stuff is a bit confusing for beginners, especially when you come from windows. Love your videos, thank you.

[Reply](#) [React](#) ...



Average Linux User

1 day ago [↩ davidA](#)

*So you just mount to "/mnt/system/" in order to address sda1, but it will still sit on ("/") in my system, right?*

Right

[Reply](#) [React](#) ...

powered by [just-comments.com](#)



Never miss a **post** from us, subscribe to our newsletter

Email

[Subscribe](#)

Explore →

[linux applications \(28\)](#)

[customization \(9\)](#)

[desktops \(11\)](#)

[xfce \(5\)](#)

[distros \(22\)](#)

[installation guide \(8\)](#)

[linux mint \(6\)](#)

[arch linux \(10\)](#)

[cinnamon \(2\)](#)

[debian \(8\)](#)

[things to do after install \(11\)](#)

[manjaro \(4\)](#)

[gnome \(3\)](#)

[ubuntu \(4\)](#)

[plasma 5 \(5\)](#)

[reviews \(6\)](#)

[kde neon \(2\)](#)

[linux general \(11\)](#)

[linux command line \(6\)](#)

[desktop themes and icons \(2\)](#)

[budgie \(3\)](#)

[solus \(3\)](#)

[server \(1\)](#)

[linux administrator \(3\)](#)

[windows \(2\)](#)

[linux general \(1\)](#)

[elementaryos \(2\)](#)

[review \(2\)](#)



[Disclaimer](#) [Privacy Policy](#)

Copyright © 2019 Average Linux User

Built with [Jekyll](#) using [Mediumish Theme](#)