

# Introduction to the OverlayFS

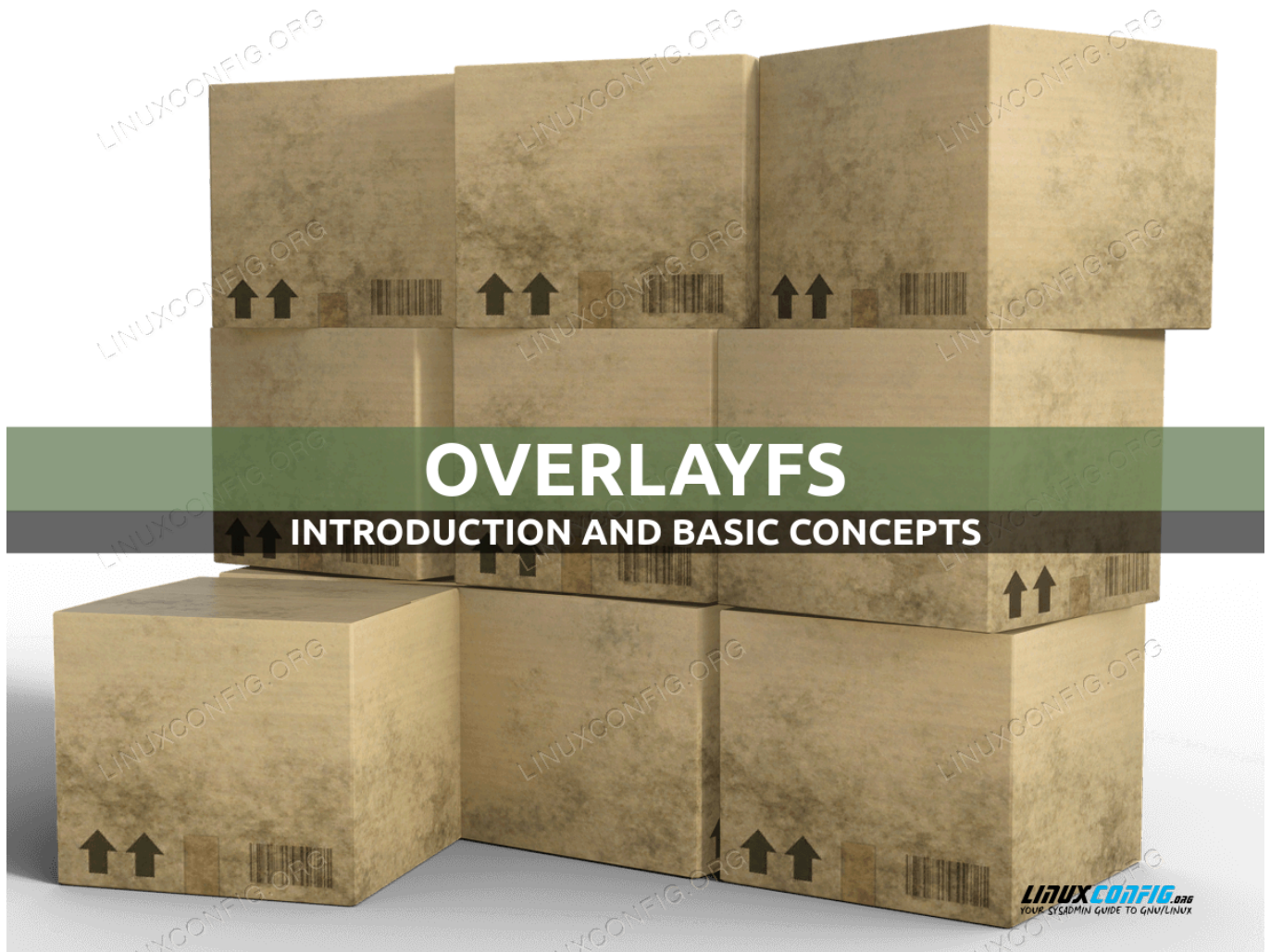
5 September 2022 by Egidio Docile

The OverlayFS pseudo-filesystem was first included in the Linux kernel 3.18 release: it allows us to combine two directory trees or filesystems (an “upper” and a “lower one”) in a way that is completely transparent to the user, which is able to access files and directories on the “merged” layer just like he would do on a standard filesystem.

In this tutorial we learn the basic concepts behind OverlayFS, and we see a demonstration of its usage.

## In this tutorial you will learn:

- The OverlayFS basic concepts
- How to combine two filesystems using OverlayFS



## Introduction to the OverlayFS

*Software requirements and conventions used*

Category	Requirements, Conventions or Software Version Used
System	Distribution independent
Software	No special software needed
Other	Administrative privileges to mount filesystems
Conventions	<p># – requires given <a href="#">linux-commands</a> to be executed with root privileges either directly as a root user or by use of <code>sudo</code> command</p> <p>\$ – requires given <a href="#">linux-commands</a> to be executed as a regular non-privileged user</p>

## Introduction

We all should be familiar with the standard behavior the Linux kernel adopts when mounting a filesystem: files and directories existing in the directory used as mountpoint are masked, and become unavailable to the user, while those existing on

the mounted filesystem are displayed. The original files can be accessed again only when the filesystem is unmounted. This also happens when we mount multiple filesystems on the same directory. When the OverlayFS pseudo-filesystem is used, instead, files existing on the different layers are **combined**, and the resulting filesystem can be mounted itself.

---

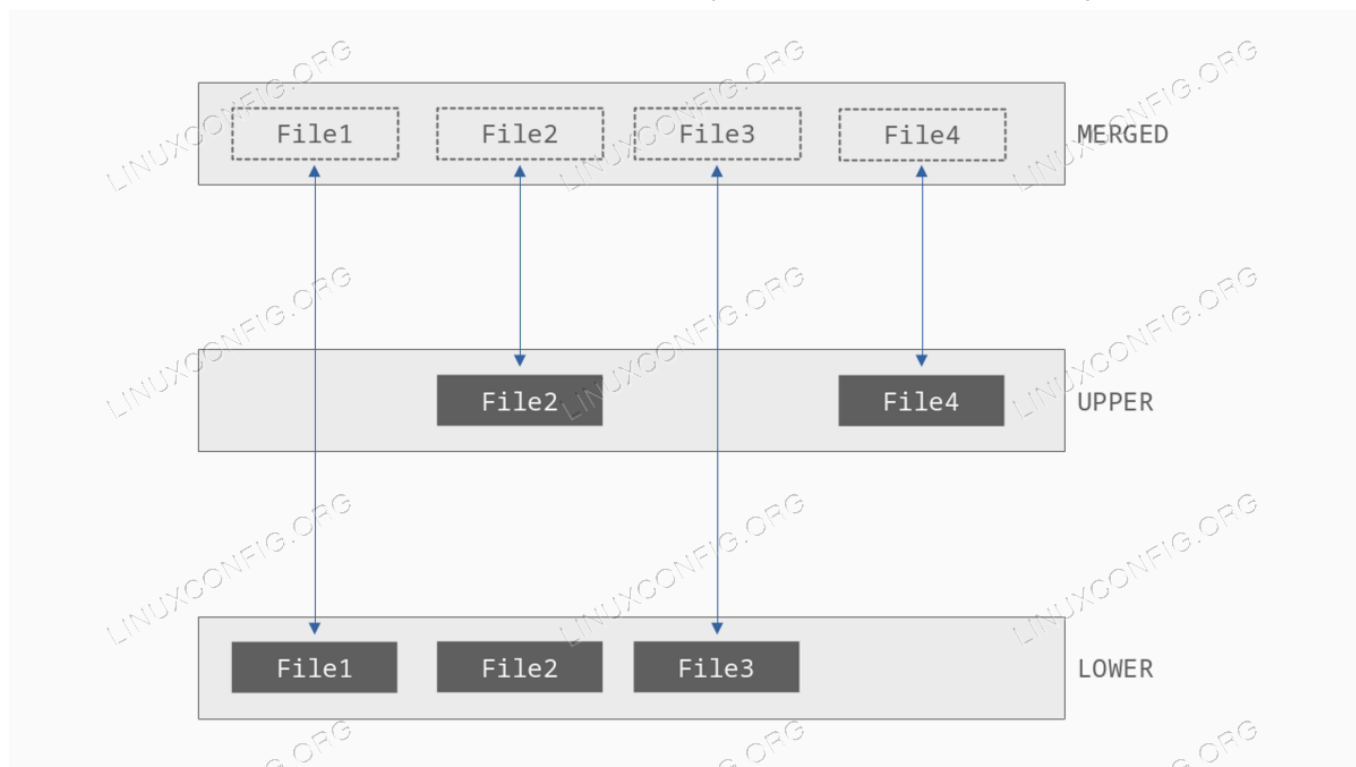
The OverlayFS is typically used on systems running on embed devices, like [OpenWRT](#), where is useful to preserve a basic set of configurations and at the same time allowing the user to perform modifications. OverlayFS is also at the base of the “overlay” and “overlay2” Docker storage drivers. Let’s see what is the main logic behind it.

## OverlayFS: basic concepts

Two layers are involved in the working of the OverlayFS: a lower and an upper one. The lower layer is typically mounted in **read only mode**. When this setup is used, since no direct changes to files and directories hosted on it are possible, it can be used as a safe fallback. The upper layer, instead, can be mounted in read/write mode. The file existing on the two layers are combined and become accessible in what we can call the “merged” layer, which can be mounted itself as a standard filesystem.

As we can observe in the picture below, a file existing in the lower layer is “obscured” or “masked” by a file with the same name existing in the upper one. When the user modifies a file which belongs to the former, a copy of it is created in the latter, writable layer (this strategy is called: “copy-up”): this file masks the original one, in a process which is transparent to the user. A similar thing happens for directories, which are **merged**, instead:





The diagram illustrates how the OverlayFS works

What happens when **delete** a file or a directory? If the file we delete belongs to the upper layer, it is deleted in place; if it belongs to the lower layer, instead, the deletion is simulated by using a **whiteout file** (or an opaque directory – a directory with the **trusted.overlay.opaque** extended attribute), which is created in the writable layer and obscures the original element.

The OverlayFS is the base of the Docker **overlay** and **overlay2** drivers. On such implementation the lower, read-only layer is represented by **images**; the writable, upper layer, instead, is represented by the **containers** based on them. Images are immutable: all changes happen inside containers and are lost when containers are deleted (that is why volumes are used for persistence):



## OverlayFS usage

Let's see how to use the OverlayFS. For the sake of this example I will assume we want to combine two filesystems: the lower one existing on the `/dev/sda1` partition, and the one to be used in read-write mode to be on the `/dev/sda2` partition. The first thing we want to do, is to create the directories which we will use as mount points:

```
$ sudo mkdir /lower /overlay
```

Now, let's mount the `/dev/sda1` filesystem on the `/lower` directory, in read-only mode:

```
$ sudo mount -o ro /dev/sda1 /lower
```

The `ls` command reveals the filesystem contains just one file:

```
$ ls -l /lower
total 20
-rw-r--r--. 1 root root 23 Sep 1 10:43 file1.txt
drwx-----. 2 root root 16384 Sep 1 10:40 lost+found
```

The files contain just one line:

```
$ cat /lower/file1.txt
this is the first line
```

Now, let's proceed further. As a next step, we mount the `/dev/sda2` filesystem on the `/overlay` directory:

```
$ sudo mount /dev/sda2 /overlay
```

Once the filesystem is mounted, we create two directories on it: `upper` and `work` . The former will host the files which are part of the upper layer, the latter will be used internally to prepare files when they are switched from one layer to the other: **it must be empty and on the same filesystem of the upper one:**

```
$ sudo mkdir /overlay/{upper,work}
```

Now we can “assemble” and mount the overlay. To accomplish the task, we use the following command:

(The next code fragment has been modified for the PDF by a third party, Martin.Vahi@softfl.com, by changing a single line command to a multiline command.)

```
$ sudo mount overlay -t overlay \  
-o lowerdir=/lower,upperdir=/overlay/upper,workdir=/overlay/work \  
/media
```

We invoked `mount` passing “overlay” as argument to the `-t` option (short for `--types` ), thus specifying the type of filesystem we want to mount (a pseudo filesystem in this case), then we used `-o` flag to list the mount options: in this case “lowerdir”, “upperdir” and “workdir” to specify: the directory on which the read-only filesystem is mounted, the directory hosting the files of the upper, writable layer, and the location of the “work” directory, respectively. Finally we specified the mountpoint for the “merged” filesystem: `/media` , in this case.

We can see a summary of the overlay setup using the `mount` command:



(The next code fragment has been modified for the PDF by a third party, Martin.Vahi@softfl.com, by changing a single line command line output to a multiline text.)

```
$ mount | grep -i overlay
overlay on /media type overlay (rw,relatime,seclabel,
    lowerdir=/lower,
    upperdir=/overlay/upper,
    workdir=/overlay/work)
```

If we list the files in the `/media` directory we see only the `file1.txt` exists, which, as we know, belongs to the lower layer:

```
$ ls -l /media
total 20
-rw-r--r--. 1 root root 23 Sep 1 10:43 file1.txt
drwx-----. 2 root root 16384 Sep 1 10:40 lost+found
```

Now let's try to add a line to the file and see what happens:

```
$ echo "this is the second line" | sudo tee -a /media/file1.txt
```

If we inspect the content of the file, we can see the line was added successfully:

```
$ cat /media/file1.txt
this is the first line
this is the second line
```

The original `file1.txt` file, however, was not altered:

```
$ cat /lower/file1.txt
this is the first line
```

Instead, a file with the same name was added in the upper layer:

```
$ ls -l /overlay/upper
```





```
-rw-r--r--. 1 root root 47 Sep 1 14:36 file1.txt
```

Now, if we delete the `/media/file1.txt` file, `/overlay/upper/file1.txt` will become a whiteout file:

```
$ sudo rm /media/file1.txt
$ ls -l /overlay/upper
c------. 2 root root 0, 0 Sep 1 14:45 file1.txt
```

As stated in the [official documentation](#), a whiteout file is a **character device** (this is reflected in the output of `ls` – see the highlighted leading “c”) with the 0/0 device number.

## Conclusions

In this tutorial we talked about the OverlayFS: we learned the basic concepts behind its usage, and we saw how it can be used to combine two filesystems or directory trees, and what are some of its possible use cases. Finally, we saw how actually create an OverlayFS setup on Linux.

## Related Linux Tutorials:

- [How to print directory tree using Linux](#)
- [Debian pinning how-to](#)
- [How to install Linux kernel headers on Raspberry Pi](#)
- [Access and modify virtual machines disk images with...](#)
- [Ubuntu 22.04 Features and Release Date](#)
- [Mint 20: Better Than Ubuntu and Microsoft Windows?](#)
- [How to build an initramfs using Dracut on Linux](#)
- [Hung Linux System? How to Escape to the Command Line and...](#)
- [Introductory tutorial to Git on Linux](#)
- [An Introduction to Linux Automation, Tools and Techniques](#)

📁 Guides, System Administration

🔖 filesystem, mount, overlay

< [How to build Super Mario 64 natively on Linux](#)





› [How to send desktop notifications using notify-send](#)

## Comments and Discussions



[Start Discussion](#)

0 replies

### NEWSLETTER

Subscribe to Linux Career Newsletter to receive latest news, jobs, career advice and featured configuration tutorials.

[SUBSCRIBE](#)

### WRITE FOR US

LinuxConfig is looking for a technical writer(s) geared towards GNU/Linux and FLOSS technologies. Your articles will feature various GNU/Linux configuration tutorials and FLOSS technologies used in combination with GNU/Linux operating system.

When writing your articles you will be expected to be able to keep up with a technological advancement regarding the above mentioned technical area of expertise. You will work independently and be able to produce at minimum 2 technical articles a month.

[APPLY NOW](#)



### TAGS

18.04 **administration** [apache](#) [applications](#) [backup](#) **bash** **beginner** [browser](#)

[centos](#) [centos8](#) **commands** [database](#) [debian](#) **desktop** **development** [docker](#) [error](#)

[fedora](#) **filesystem** [firewall](#) [gaming](#) [gnome](#) [Hardware](#) **installation** [java](#) [kali](#) **multimedia**

**networking** [nvidia](#) **programming** [python](#) [raspberrypi](#) [redhat](#) [rhel8](#) **scripting** **security**

**server** [ssh](#) [storage](#) **terminal** **ubuntu** **ubuntu 20.04** [virtualization](#) [webapp](#) [webserver](#)

ABOUT US

## FEATURED TUTORIALS

VIM tutorial for beginners

How to install the NVIDIA drivers on Ubuntu 20.04 Focal Fossa Linux

Bash Scripting Tutorial for Beginners

How to check CentOS version

How to find my IP address on Ubuntu 20.04 Focal Fossa Linux

Ubuntu 20.04 Remote Desktop Access from Windows 10

Howto mount USB drive in Linux

How to install missing ifconfig command on Debian Linux

AMD Radeon Ubuntu 20.04 Driver Installation

Ubuntu Static IP configuration

How to use bash array in a shell script

Linux IP forwarding – How to Disable/Enable

How to install Tweak Tool on Ubuntu 20.04 LTS Focal Fossa Linux

How to enable/disable firewall on Ubuntu 18.04 Bionic Beaver Linux

Netplan static IP on Ubuntu configuration

How to change from default to alternative Python version on Debian Linux

Set Kali root password and enable root login

How to Install Adobe Acrobat Reader on Ubuntu 20.04 Focal Fossa Linux

How to install the NVIDIA drivers on Ubuntu 18.04 Bionic Beaver Linux



[How to check NVIDIA driver version on your Linux system](#)

[Nvidia RTX 3080 Ethereum Hashrate and Mining Overclock settings on HiveOS Linux](#)

## LATEST TUTORIALS

[Changing Your Default CLI Text Editor in Ubuntu/Debian](#)

[Automate and Personalize: Change Ubuntu Desktop Wallpaper via Command Line and Cron](#)

[Installing Docker on CentOS 9 Stream](#)

[Installing AWS CLI on Ubuntu: A Step-by-Step Guide](#)

[How to install, configure and customize Waybar on Linux](#)

[Resetting Root Password on Ubuntu 24.04 Linux](#)

[Introduction to Polkit: Navigating Authorization Frameworks in Linux](#)

[Guide to Installing XFCE Desktop on Ubuntu Linux](#)

[Guide to Installing Budgie Desktop on Ubuntu Linux](#)

[Guide to Installing XFCE Desktop on Debian GNU/Linux](#)

[Guide to Installing Cinnamon Desktop on Ubuntu](#)

[How to turn a Raspberry Pi into a surveillance system with ZoneMinder](#)

[Guide to Installing Mate Desktop on Ubuntu](#)

[How to configure Certificate Authority on Ubuntu/Debian](#)

[Step-by-Step Guide: Adding Certificates to Ubuntu's Trusted Authorities](#)

[How to check GPIO status](#)

[Raspberry Pi as Music Streamer](#)

[10 Essential Methods to Reboot Your Ubuntu Server](#)

[How to create GNOME dynamic wallpapers](#)

[How to create optimized virtual machines with Quickemu on Linux](#)

