Deploying Linux by Cloning Method

Restoring a Cloned Distribution to Multiple Computers with Rescuezilla

Table of contents

Introduction

This manual is intended for users who are familiar with Linux and at least partially comfortable using the command line.

My advice: read this entire manual before starting, and ensure you understand everything.

Once a Linux system is installed on a computer, it can be moved to another machine without compromising its ability to function.

Two conditions are necessary for this to work:

The intended architecture must be compatible: for example, x86_64 (also known as Amd64 or 64-bit) works only on PCs equipped with a 64-bit processor.

A 32-bit OS can run on both 32-bit and 64-bit processors. 32-bit systems are lighter on resources, but they recognize less than 4 GB of RAM (system memory).

The graphics card or chipset of the computer must not be too "exotic".

In summary, it works in most cases, thanks to the many drivers built into the Linux kernel designed for all possible kinds of hardware.

Installation Modes

Classic mode: Using a CD-ROM, a DVD, or a USB stick prepared to boot live systems (nomad or demonstration mode).

Alternative modes: Installation via PXE server (network), disk-to-disk copies, partition copying with Gparted (both methods are slow), and finally, cloning of installed and customized systems followed by restoration to one or more computers.

Installation from a Clone

Objective

To maintain a homogeneous fleet and save time during deployment operations. Restoring a cloned Linux system can take between 5 and 10 minutes, depending on the system size and the performance of the computer on which Rescuezilla is used.

This represents a significant time saving compared to classic methods.

Furthermore, the clone can be created from a physical computer or a virtualized system (using VirtualBox, for example).

It is therefore not mandatory to tie up a physical computer to create clones, and the virtual machine can be reused as many times as needed as updates are released, including by restoring the clone to a new virtual machine if desired.

Scenario

Suppose you have installed the operating system of your choice on a powerful computer (always to save time).

Let us assume it is not Windows : if it were Windows, you would need target machines that are identical in brand and model to the one where the system was installed, which is not the subject here.

Once your system is installed, you will have added or removed programs for the users, perhaps modified its appearance ;

in short, you will have prepared it for deployment in the context that suits you (administrative use, developer use, or family use).

Thanks to this preparation, cloning this ready-to-use system will save you time on post-installation tasks. Subsequently, each user only needs to create their personal account using the dedicated tool from the system section of the application menus, which is very fast to do.

Order of Operations

1. Download the latest version of Rescuezilla

Rescuezilla is a fork of Clonezilla offering a graphical interface for common needs, provided as a Linux distribution. It is maintained by an American developer on an Ubuntu base.

In your Linux system, open a terminal in the desired directory (for example /home/user/ISOS) and run:

```
wget -c https://github.com/rescuezilla/rescuezilla/releases/download/2.6/rescuezilla-2.6-64bit.oracular.iso
```

If you prefer curl:

```
curl -L -O https://github.com/rescuezilla/rescuezilla/releases/download/2.6/rescuezilla-2.6-64bit.oracular.iso
```

2. Prepare your bootable media

We most often use a USB stick, as optical drives are becoming less common, and Linux distributions have grown beyond the 700 MB capacity of a CD-ROM.

Take a blank or empty USB stick and follow the Ventoy method: https://ventoy.net/en/index.html – this allows you to use the entire space of the medium (USB stick or external USB storage), as it enables booting directly from ISO disk images.

The instructions for installing Ventoy from various operating systems are on the site. Under Linux, download the Linux archive, extract it, navigate to the extracted folder, and run the Ventoy2Disk.sh script (read the README first).

Precautions:

- Plug in the target USB stick.
- Open a terminal and type ***sudo blkid | grep sd*** : this will allow you to verify the name of your USB stick (Don’t install Ventoy on the disk running your host system !)
- Copy Rescuezilla into the Ventoy USB stick. The use of the following command is recommended : ***rsync <source> <destination> && sync***

or, if you use copy-paste, take the precaution of verifying the md5 or sha512 hash once the ISO is at the destination.

Other tools for preparing a bootable USB stick:

- **Balena Etcher** (I don’t know it)
- **Gnome-Multi-Writer** is a graphical tool available in Debian and Ubuntu repositories.

It allows installing a Linux system on multiple USB sticks simultaneously.

Suited to small USB sticks: it uses the entire space by copying the ISO bit by bit, so the USB stick will be in ISO 9660 format.

- Its use is particularly suited for quickly preparing several bootable USB sticks. To reuse them afterwards, they will need to undergo a low-level format (for example, creating a new partition table with Gparted).

3. **Install the clone onto a USB medium**

The preparation of the clone itself is not covered here; you will discover it easily by booting into the Rescuezilla live environment from a computer you wish to clone.

To install a clone that is already ready for deployment: you will need to access the clone of the distribution, so make a copy from its original location.

The use of rsync is strongly recommended: this tool performs copies by synchronization. It is powerful for copying data over the network (if you use this option for system dumps and restorations) and will also work perfectly when using USB or other devices connected directly to the computer.

The following command copies the source file in verbose mode while displaying progress. Once finished, the && sync forces data synchronization to disk, and allows you, when the command returns the prompt, to confirm that the copy is truly complete:

```
rsync -v --progress <path to source file> <path to destination> && sync
```

> *If you need to preserve file attributes, use rsync -av instead of rsync -v (though this will make no difference if your USB medium contains an NTFS filesystem, which does not preserve permissions). As for FAT32, do not use it in this context: the file size limitation could cause the operation to fail.*

(Sometimes, the process might remain pending once it reaches 100%, and it would seem to no longer move; this tip allows to resolve this).

Steps:

- Boot the target computer (where you want to install the clone) from your Rescuezilla USB stick (choose the language at startup). Once on the desktop, you can close the Rescuezilla window (which launches automatically at startup) and check the presence and characteristics of the hard drive or SSD using Gparted (and possibly Gsmartcontrol, which can be installed with apt commands).
- Start Rescuezilla, select the restore options, and plug in the medium containing the clone files.
- If needed, consult the Rescuezilla documentation : https://rescuezilla.com/screenshots and https://rescuezilla.com/help. (Always install partitions or the entire disk clone at the beginning of the target disk, you will see why at the end of this manual).
- The destination disk should have a partition table and one or more formatted partitions in order to be recognized by Rescuezilla.
- Start the cloning process, and once it is finished, note how long the operation took.

4. Post-installation before restarting

Once cloning is complete, two elements remain to be considered: the clone was likely smaller in disk size than the destination disk.

- Expand the partition

This is the moment to close the Rescuezilla window and launch or relaunch Gparted: to expand the last partition to the right, in order to take advantage of the full disk space. This is generally the partition dedicated to /home, unless the entire system resides in a single partition.

Note: partitions cannot be expanded to the left. They can only be moved to the left, which requires a great deal of time, as the data is copied bit by bit.

- Changing (or not) the UUIDs

See the related feature request from 3 years ago : https://github.com/rescuezilla/rescuezilla/issues/312 (still open).

The other criterion answers the following question: after cloning onto a computer intended to be deployed, will it be placed in a fleet where other computers also installed from the same clone will be present on the same local network?

This question concerns a technical point: the UUID (Unique Identifier) of the partitions. The command sudo blkid provides the UUIDs of the partitions.

Explanation:

The clone is provided with its own UUID(s). Machines on the same local network should each have their own set of UUIDs, in order to avoid inconsistencies when mounting filesystems from one machine to another.

> Attempting to mount the filesystem of a machine with the same UUID could create confusion or mount failures. And some network services or fleet management tools might use identifiers based on UUIDs (not necessarily those of the filesystems, but other types of UUIDs generated at installation time, such as /etc/machine-id under systemd). The system would not know which machine to target.

In this case, at the end of each cloning operation, it is useful to change the UUID(s) of each computer prepared in this way.

We start by ensuring that the target partitions are properly unmounted (with administration mode)

```
umount -a
```

To change the UUIDs for ext2/3/4 filesystems (the most common for desktop computers and laptops):

```
tune2fs -U random /dev/<device>
```

where /dev/<device> can be /dev/sda1, /dev/sda2, /dev/sdb1, /dev/nvme0n1p1, etc.

For servers using the XFS filesystem:

```
xfs_admin -U generate /dev/<device>
```

For BTRFS:

```
btrfstune -u /dev/<device>
```

Then note the new UUID(s) of the partitions.

You will need to perform a **chroot** on the system partitions (consult this documentation if needed: https://doc.ubuntu-fr.org/chroot) for the following steps.

- Once you are positioned in the chroot through the terminal, using your favorite text editor (it will be nano by default, but you can install another text-mode editor if you prefer), modify the UUIDs accordingly for each partition in the /etc/fstab file.
- To change them in /boot/grub/grub.cfg, launch the command "grub-mkconfig -o /boot/grub/grub.cfg"

If you do not feel confident editing /etc/fstab in administrator mode, make a backup copy of this file before starting its edition.

- For the machine identifier, with modern Linux systems using SystemD: you will launch the following command: "truncate -s 0 /etc/machine-id && systemd-machine-id-setup".

Remember to save your modifications. Quit the chroot, and unmount in reverse order if you wish to perform other operations in the live environment before restarting.

- Restart the computer, remove the bootable USB stick and the device containing the clone at the end of the shutdown, and check if the computer starts well on its new system.