

# Installing Linux

There are [many different Linux distributions](#), which all come with advantages and disadvantages. There is not really a blanket solution for which system you should choose. However, the best way to get into it is to just start, and learn along the way.

For this guide, we recommend installing [DietPi](#). Originally developed as an operating system for the Raspberry Pi, it now supports most Single Board Computers, as well as native PCs. It's very light-weight, meaning that it doesn't take much space and doesn't use a lot of resources like processing power and RAM, ensuring that it will run fine even on slower machines. Since we are recommending to use [docker](#), which works a little bit like a small operating system in a box and doesn't have many requirements outside of that, we generally want our operating system to take up as little resources as possible. If you prefer to use another Linux distribution, feel free to skip this section and continue at [setting\\_up\\_ssh](#).

## 1.1 Installing DietPi

In order to install DietPi you need a few things:

- A **working computer** with internet access (as in: a computer to work from (also one that works)) from which to prepare the installation
- A (non-Apple) **target computer** on which you want to install DietPi. For this guide, we will also assume that you have access to a screen and a keyboard as well (if you are using an old laptop, the internal screen and keyboard suffice). You can read more about hardware choices [here](#). - An empty USB drive of minimum 2 GB capacity (to install from) For now, this guide will not include a tutorial on how to install Linux on an old Apple computer. You can read more about this [here](#), but the *tl;dr*; is that Apple has made it increasingly difficult to install anything except macOS on their machines. If an old Mac is all you got, there are many tutorial such as [this](#) one that you can. Ultimately, your choice in operating system should not change the process in the further guides dramatically.

### Downloading DietPi for single board coimputers

There are [many variations](#) of DietPi available on the official website, depending on what kind of machine you are planning to install it on. There are versions for most common [SBCs \(Single Board Computers\)](#), which is one of the main advantages of DietPi. If you are using an SBC, find the appropriate version of DietPi on their website and download it.

### Downloading DietPi for a native (non-SBC) PC

If you plan to install DietPi on a *regular* (non-SBC) computer, like an old laptop, you first need to find out whether it's BIOS based or UEFI based. It's not really important to understand what this means in order to continue, it's just about downloading the correct installer.

## Distinguishing UEFI/BIOS on Windows 11+

Basically, if your computer is newer, chances are bigger that it's UEFI based. The older it is, the more likely it's BIOS based. If it runs Windows 11, you are running UEFI for sure since it doesn't support BIOS.

## Distinguishing UEFI/BIOS on Windows 10 and older

If the computer you want to use is running a version of **Windows** older than Windows 11, you can check whether your machine is UEFI or BIOS based by pressing **Win + R** to open the Windows Run dialog, type in the command `msinfo32`, and then press Enter. This will open the System Information window. Find the entry **BIOS mode**. If it's set to UEFI you are using UEFI, if it's set to Legacy you are using BIOS.

## Distinguishing UEFI/BIOS on Linux

If you are already running Linux, the easiest way to find out if you are running UEFI or BIOS is to check for a folder called `/sys/firmware/efi`. One way to do this is to open a [terminal](#) and to enter the command `ls /sys/firmware/efi`. If you are using BIOS, the folder will be missing. If the folder exists, you are running UEFI.

Once you know whether your computer is UEFI based or BIOS based, download the corresponding version from the [official website](#) to your **working computer**.

## Extracting the image

The file you downloaded in one of the previous steps is packaged in a 7z archive, which you need to extract first. Functionally, this is similar to a .zip or .rar file. You might need to download additional software in order to extract the archive.

## Extracting 7z on Windows or macOS

If your **working computer** runs **Windows** you can use [7-Zip](#).

If your **working computer** runs **macOS** you can use [The Unarchiver](#).

## Extracting 7z on Linux

If your **working computer** runs **Linux** you can use p7zip, the command line version of 7-Zip. On Debian and Ubuntu-based systems, you can simply open a [terminal](#) and type:

```
$ sudo apt install p7zip
```

Once p7zip is installed, navigate to the downloaded file and type the following to extract it:

```
$ 7zr x PUTYOURFILENAMEHERE
```

Replace **PUTYOURFILENAMEHERE** with the correct name of the downloaded archive, e.g. **DietPi\_RPi-ARMv6-Bullseye.7z**. This will extract the DietPi image file for you to use.

After all of this, you should be left with an *image file*. If you don't know what this means, just make sure you have a file that ends with `.iso`.

## Flashing the Image

We now want to take this image file, and put it on a flash drive like a USB stick or SD card, which we can then use to install Linux. This process is called *flashing*.

## Flashing if your target computer is a single board computer

There are several tools to do this. If your **target computer** is a single board computer, or a native PC that's BIOS-based, the easiest way is probably [balenaEtcher](#), which is free and available for Windows, macOS and Linux.

It's pretty straight forward: after downloading balenaEtcher to your **working computer**, plug in your flash drive and open balenaEtcher. Select **Flash** from **File**, then locate the installer image you extracted in the previous step (it should be an `.iso` file). Then select your flash drive as a target and click **Flash**!

This will erase all the data on the selected target. Double check if you have selected the correct target drive and if there are **\*\*really\*\*** no files on there that you don't want to lose.

## Flashing if your target computer is a native PC

If your target computer is a BIOS-based native PC, you can use balenaEtcher as explained in the previous step. However, if your target computer is UEFI-based, things are slightly more complicated.

## Flashing for a native PC on Windows

If your working computer is running **Windows**, you can use the free tool [Rufus](#). There is a portable version of Rufus, which you can download and run directly without installing it first. Then plug in your flash drive, and follow these steps:

- Select the USB device
- Select the downloaded **DietPi** image
- Select **GPT** as partition scheme
- Select **UEFI** as target system
- Click on **Start** button

⚠ This will erase all the data on the selected target. Double check if you have selected the correct target drive and if there are **\*\*really\*\*** no files on there that you don't want to lose.

## Flashing for a native PC on Linux or macOS

Sadly, if your **working computer** runs **Linux** or **macOS**, things get a little bit more difficult. For reasons that are related to the distinction between BIOS and UEFI, balenaEtcher is not a good tool for flashing the USB stick for a UEFI-based native PC and the DietPi Instructions specifically advise against using it. However, the recommended alternative [Rufus](#) is only available for Windows. This is not a problem *per se*, but it will require us to start using the [terminal](#).

The first thing we need to do is identify and format the USB stick from the terminal. Start by opening a new terminal window. This part differs slightly from macOS to Linux:

### Identifying and flashing your flash drive on Linux

On **Linux**, use the command `lsblk` to list all *block devices*. It's a bit complicated to explain what a block device is, so we will not go into it, and it's not important for following this guide. This will present you with a list of all block devices connected to your computer. It might look a bit confusing at first, but don't worry.

Most likely you are probably looking for a device that's called `sda`, `sdb`, `sdc` or something like that. The easiest way to identify your USB stick in this list is probably through its size.

Let's illustrate this on an example. We are looking for our USB stick, which we know has a **capacity of 2.5GB**. After typing in `lsblk` we might receive output that looks something like this:

```
NAME      MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
...
sda        8:0    0 232.9G  0 disk
├─sda1     8:1    0  975.3M  0 part  /boot
├─sda2     8:2    0   15.3G  0 part  [SWAP]
├─sda3     8:3    0   36.4G  0 part  /
└─sda4     8:4    0 180.3G  0 part  /home
sdb        8:16   0    2.5G  0 disk
└─sdb1     8:17   0    762M  0 part
...
```

There might be more entries in this list, depending on your configuration. We see here that there is a device called `sda` that has a capacity of 232GB, and a device called `sdb` with a **capacity of 2.5GB**. This tells us that the name of our USB stick is `sdb`, which means the *device path* is `dev/sdb`

Be careful when identifying your USB stick, you don't want to mess this up as you might end up formatting the wrong disk later. If you're unsure, unplug your USB stick and run the command again. By comparing the two outputs, you should be able to tell the name of your USB stick!

> In the following steps, we will format the USB stick. This will erase all the data on the selected target. Double check if you have selected the correct target drive and if there are **\*\*really\*\*** no files on there that you don't want to lose.

Before flashing the USB drive, we need to make sure that it's in the correct [file system](#), in this case

FAT32. This is not something you need to understand when following along to this guide, however it's probably something that is useful to understand on at least a basic level in the long run! The easiest way to do this is to format the drive. First, we will use the `fdisk` command to delete all the data on the device.

```
sudo fdisk [DEVICEPATH]
```

Replace `[DEVICEPATH]` with the device path you found in the previous step, for example `/dev/sdb`. You will enter the `fdisk` menu. From here:

- Type `d` to proceed to delete a partition
- Type `1` to select the 1st partition and press enter
- If your USB stick had multiple partitions, type `d` to proceed to delete another partition (`fdisk` should automatically select the second partition)
- Type `n` to make a new partition
- Type `p` to make this partition primary and press enter
- Type `1` to make this the first partition and then press enter
- Press enter to accept the default first cylinder
- Press enter again to accept the default last cylinder
- Press `t` to set the partition type
- Press `1` to select the first partition
- Type `0c` (zero, not O!), which stands for FAT32
- Press enter to confirm
- Press `a` to toggle the boot flag
- Press `1` to select the first partition
- Type `w` to write the new partition information to the USB key

This last step should close the `fdisk` menu. At this point, we just need to unmount the device using `umount`, and create the file system using `mkfs`. To do this, enter these two commands (Replace `[DEVICEPATH]` with your *device path*, as above):

```
umount [DEVICEPATH]
sudo mkfs -t vfat [DEVICEPATH]
```

## Identifying and flashing your flash drive on macOS

The process is very similar to doing it on Linux, but we still need to write this section :)

## Installing DietPi

Once your flash drive is ...flashed, your **working computer** has done its job, and we will start working with your **target computer**.

Start by plugging your flashed flash drive into your **target computer**. If you are using an external screen and keyboard for this installation, plug them in as well. Then [boot up](#) the **target computer**.

If your **target computer** currently has a running operating system, you might be presented with a prompt asking you to select a boot device. This just means that you should select from where the computer should take the operating system that it is about to start running—in this case you should identify and (using the arrow keys) select the flash drive you just prepared, then press ENTER.

You should be presented with a neon-green dialogue asking you to select the version of DietPi you wish to install. Unless you are experiencing any issues during this step, select *Default Settings*.

Follow the on-screen instructions for the rest of the installation. It will ask you to select your keyboard layout, and the source for the installation (you should only see one possible source if you followed this guide up to here—it should start with DietPi\_ and continue with the version of DietPi you downloaded in the beginning of the tutorial).

Next you will be asked to select the hard drive that you want to install DietPi on. In many cases you will only see one option here, in case your target computer has multiple hard drives installed, pick the most [appropriate](#) one.

⚠ This will erase all the data on the selected hard drive, so this is a last warning to be sure that you actually want to erase this drive!

After this, the installation should begin. This might take some time. When it's done, the system should shut down by itself.

## Running DietPi for the first time

To run DietPi, simply unplug the flash drive you used for installing DietPi, and turn on the machine. Depending on your **target computer** you might have to press a button like esc, F10, F2, F12, F1 or del (this depends on the manufacturer). You need to select your USB stick as a boot device.

This part is underdeveloped and could be expanded on. The following is taken from the official [\[\[https://dietpi.com/docs/install/|DietPi Install Guide\]\]](https://dietpi.com/docs/install/) for installing the system, however it requires previous knowledge, so it would be nice to explain in more detail for newer users!

During the initial boot, the following dialog may appear to boot from the USB stick:



After booting the graphics selection dialog appears:

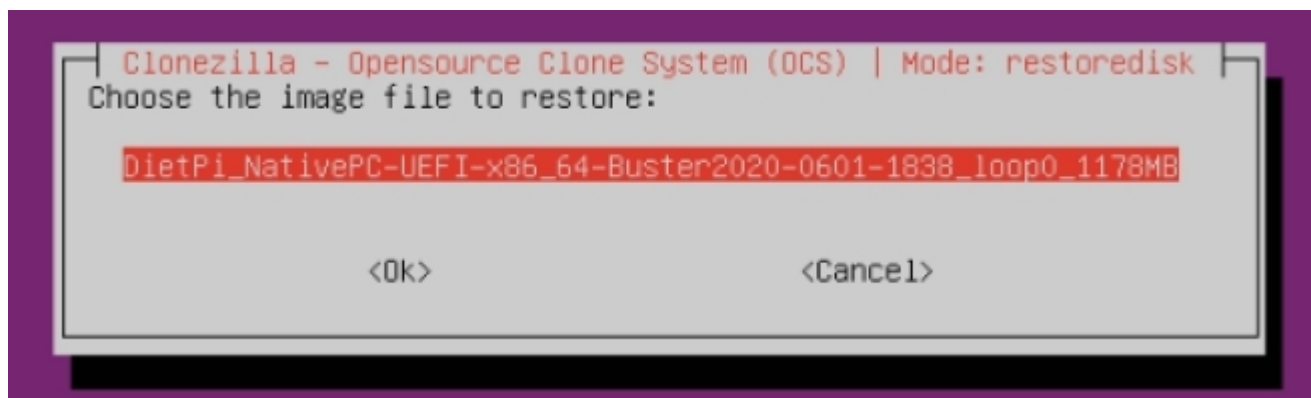


You can select the default settings. In case of problems, please select "Safe graphic settings".

Once this step is completed, you will be able to select a different keyboard. If necessary, change your keyboard settings and go through the appropriate dialogues.

Then the installation process begins with the help of the wonderful Clonezilla tool.

Select the image file to be installed on the target PC's harddisk. Normally you should only see one single option:



After this, you have to select the target PC's harddisk where your DietPi shall be installed. In this example there is only one harddisk present:



After this, the installation process starts with several steps, e.g. showing the process of the image copying:







client like [PuTTY](#) to connect from a remote system. The SSH server [Dropbear](#) is installed and enabled by default on DietPi.

- Most SBCs alternatively allow to connect a serial console via **UART**, which is by default enabled on DietPi as well.

A login prompt will appear. Use the initial credentials:

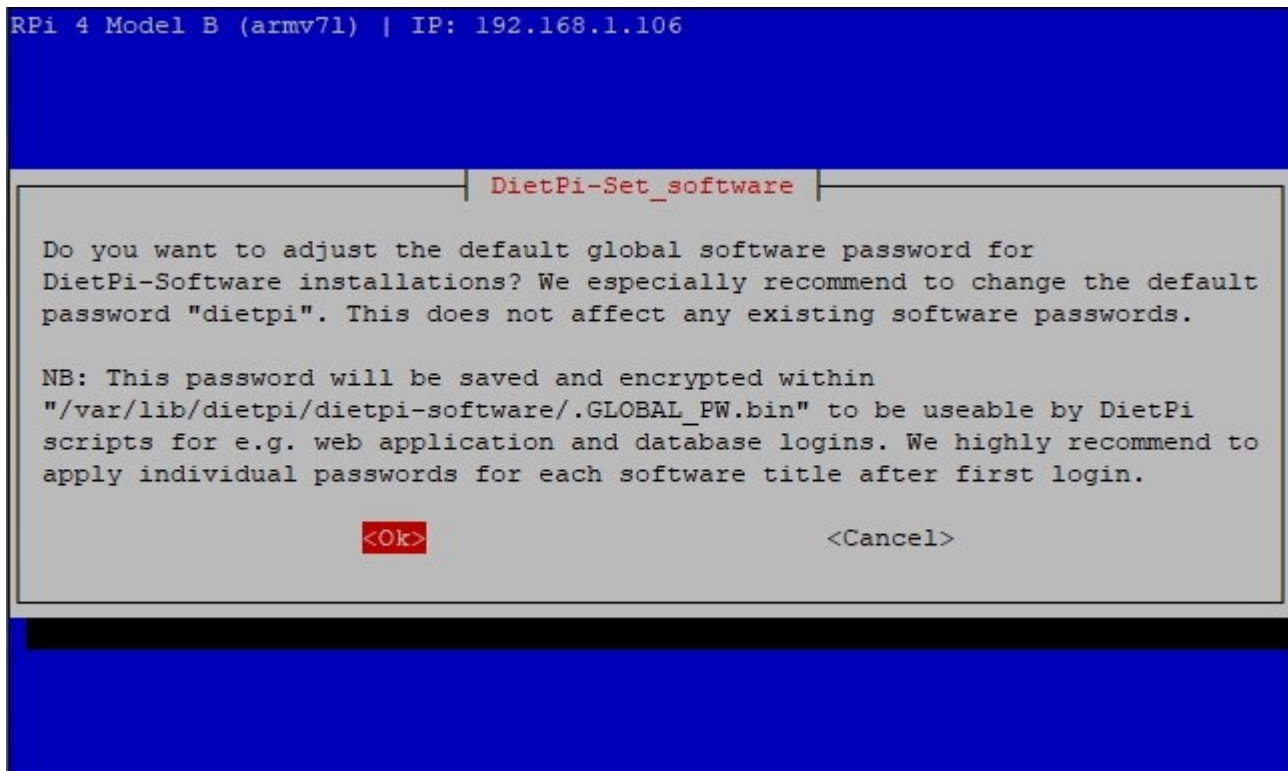
- login: root
- password: dietpi (resp. the one you set via `dietpi.txt`)

DietPi will then immediately begin to search for and install updated software packages, which will take some time to complete.

Once the packages have been updated, DietPi will ask you to confirm whether you would like to enable user analytics.

The default DietPi password is public, so you'll be asked to change this at the next stage for both the root and dietpi user accounts. Select OK and hit Enter, then provide your password (twice) to confirm.

You can change the password again later by typing `passwd` at the terminal or also via the command line script `dietpi-config` (within the "Security options").



The base installation of DietPi is minimal **by design**, allowing you to choose what software you want to install and use: Just run `dietpi-software` and install **[\\*\\*DietPi Optimised Software\\*\\*](#)**.

You can return to the **DietPi-Software** tool to make further changes at any time by typing `dietpi-software` at the terminal, or enter `dietpi-launcher` and select **DietPi-Software** tool.

If you want to make further changes to your DietPi configuration, you can run `dietpi-launcher` at the terminal to view all the available DietPi tools, including **DietPi-Update** to update your device and **DietPi-Backup** to back up your device.

For more details, check [DietPi Tools](#) section.

From:

<https://self-hosting.guide/dokuwiki/> - **Self Hosting Manual**

Permanent link:

[https://self-hosting.guide/dokuwiki/how/installing\\_linux](https://self-hosting.guide/dokuwiki/how/installing_linux)

Last update: **2022/12/09 13:55**

