

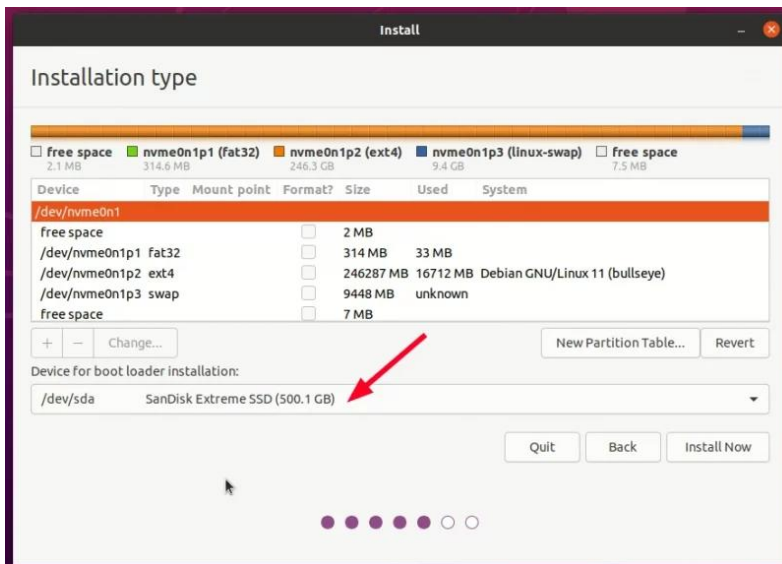
Installation of a Stand-alone Bootable Linux Build on a USB (Use as a Daily Drive or Rescue Disk)

The problem with the bootloader on UEFI systems

The one major problem with the setup is bootloader installation on UEFI systems.

A system can only have one active ESP partition at a time and this causes issues.

While installing Linux, even if you choose the USB as the destination for the bootloader, the existing ESP partition is still used for placing the EFI file for the new distribution.



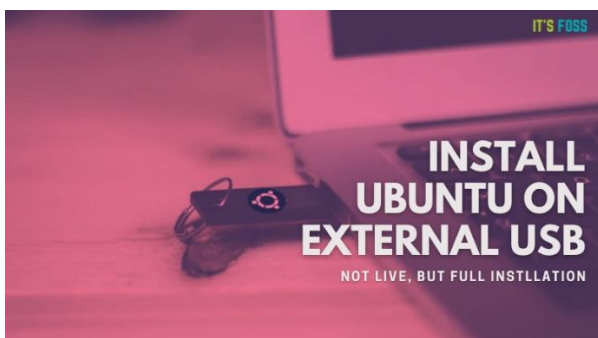
Even if you choose to install bootloader on the external USB, it doesn't work if hard disk already has an ESP partition.

This means that the Linux installed on the external USB will be the last to update and control the bootloader. This may create several problems:

- The system you used for installing Linux on USB may not boot and end up with a grub error if you don't have the Linux USB plugged in.
- The Linux USB you created won't boot on other systems because its EFI files are on the ESP partitions of the system which was used to install Linux on the USB.

As a result the USB won't boot as expected and additional steps are needed to allow the USB to boot as required.

Things to know before installing Linux on a USB



Installation of a Stand-alone Bootable Linux Build on a USB (Use as a Daily Drive or Rescue Disk)

The solution or workaround to the bootloader problem is that you do not let the installer know that there is already an ESP partition.

The first step is to remove the ESP flag from the ESP partition before installing Linux on the USB and put it back after installation. This way, you fool the Ubuntu installer into thinking that there is no existing ESP partition and it will create and use a new ESP partition on the USB. The original ESP partition on the hard disk is untouched.

Things to note:

- You'll need two USB keys. One for live USB and another one where Linux will be installed (Linux To Go).
- Use a USB of at least 4 GB for the live system and at least a 32 GB USB for the stand-alone Linux USB, Linux To Go. It is not a bad idea to ensure that the stand-alone USB is not partitioned and is blank.
- **Using USB 3.0 is highly recommend for both live USB and the actual Linux To Go USB.** USB 2.0 will work but will be significantly slower for both installing and using Linux.
- Even if you use USB 3, installing Linux will be multiple fold slower than normal Ubuntu installation. Have patience and time to spare.
- Using a Linux system from USB will always be slower than actual hard disk or SSD.
- **When you want to use the Linux USB on a system, you'll have to go to the UEFI boot settings to boot from the USB (unless the system is set to boot from USB by default).**
- **You must take account of your decision on secure boot:**
 - If you have decided to turn secure boot off there are no further consequences
 - If secure boot is on, then the Linux To Go USB you build will require that you enrol the build with a Machine Owner's Key (MOK) as part of the first boot once it has been built, if you have decided to compromise the signed state of the software in some way (typically by using third-party software in the build). As part of this you will be required to enter a password, which you will need exactly once only during the first boot to validate the MOK. If you forget the password, you will need to repeat the build process.

Installing complete Ubuntu on a flash drive

The steps are:

1. Create a live Ubuntu USB
2. Use the live USB to install Ubuntu on another USB by removing ESP partition flags before and after the installation
3. Using the Ubuntu USB on various systems
4. Getting your USBs back in normal condition

Part 1: Create a live Ubuntu USB

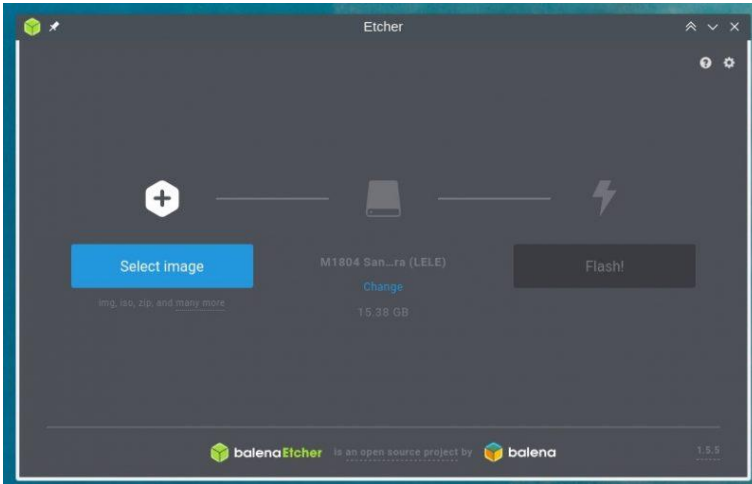
Without all the detail the process is:

Installation of a Stand-alone Bootable Linux Build on a USB (Use as a Daily Drive or Rescue Disk)

Go to Ubuntu's website and download a current Ubuntu release of your choice.

There are several tools available for writing the ISO image to the USB. If you are running Windows Rufus is a good choice, alternately you may use Etcher which is available for Linux, Windows and macOS.

For Etcher: Plug in your smaller capacity USB. Run Etcher and browse to the downloaded ISO and USB



Then hit the flash button. The install USB will be built. A separate document exists for using Rufus.

When you have created the live USB, it's time to use it to install Ubuntu on the other Linux To Go USB.

Part 2: Installing Ubuntu on the USB

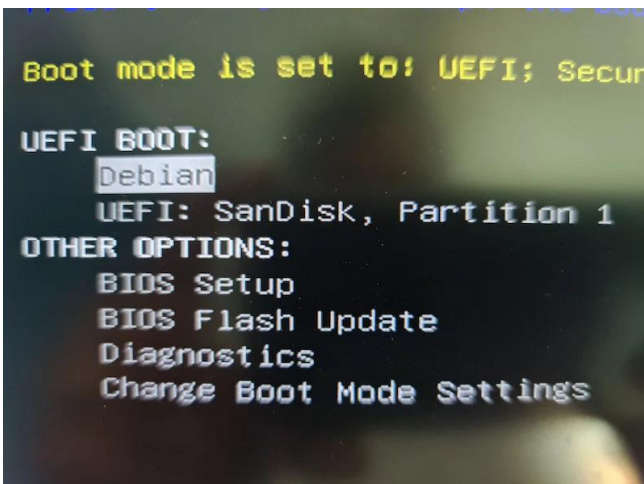
Go to the target PC where you are going to do the build. Plug both of the USB to that PC. You have to boot from this live install USB now and for that you'll have to access the boot settings.

Step 1: Boot from live USB

Plug in both USBs to your system and restart it. While the system is booting and it shows the manufacturer's logo, press the keys which give access to the BIOS/UEFI settings. Different manufacturers set different keys for accessing the BIOS settings. You will find this information in the PC documentation or on-line.

Note: Some systems won't allow booting from live USB if secure boot is on. If that's the case, [disable secure boot](#) first.

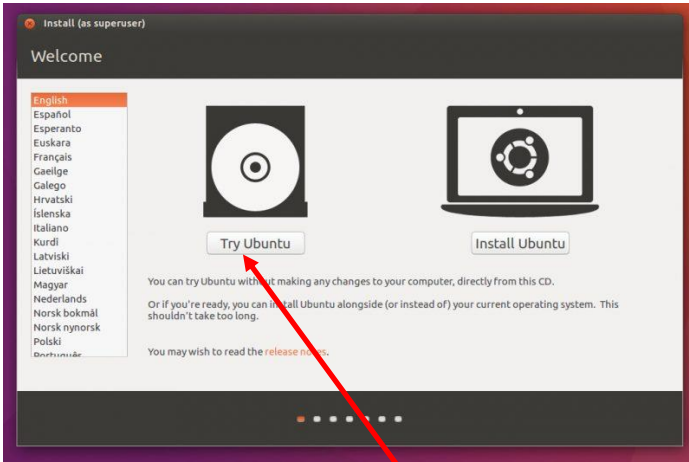
The screens below are from a Dell system and the boot settings showed this screen. Debian is installed on this system. What you will see depends on the make and model of your PC and what is installed on it.



Installation of a Stand-alone Bootable Linux Build on a USB (Use as a Daily Drive or Rescue Disk)

Select the Install USB as the boot source, then allow the system to reboot. It will boot from the Boot USB. And the Linux install process will start.

When you are in the live session, go with “Try Ubuntu” option:



Counterintuitively, click the Try Ubuntu option here

Step 2: Disable ESP flags from the ESP partition

Press the Windows key and type GParted. This will open the GParted partition manager which is (usually) already present on the Ubuntu ISO.



Start GParted in Ubuntu by double clicking it.

If it is not found, connect to the internet, and download it with the Ubuntu Software application. Alternately open a terminal, run **sudo apt update** and then install GParted on Ubuntu. You will need the Internet for this process.

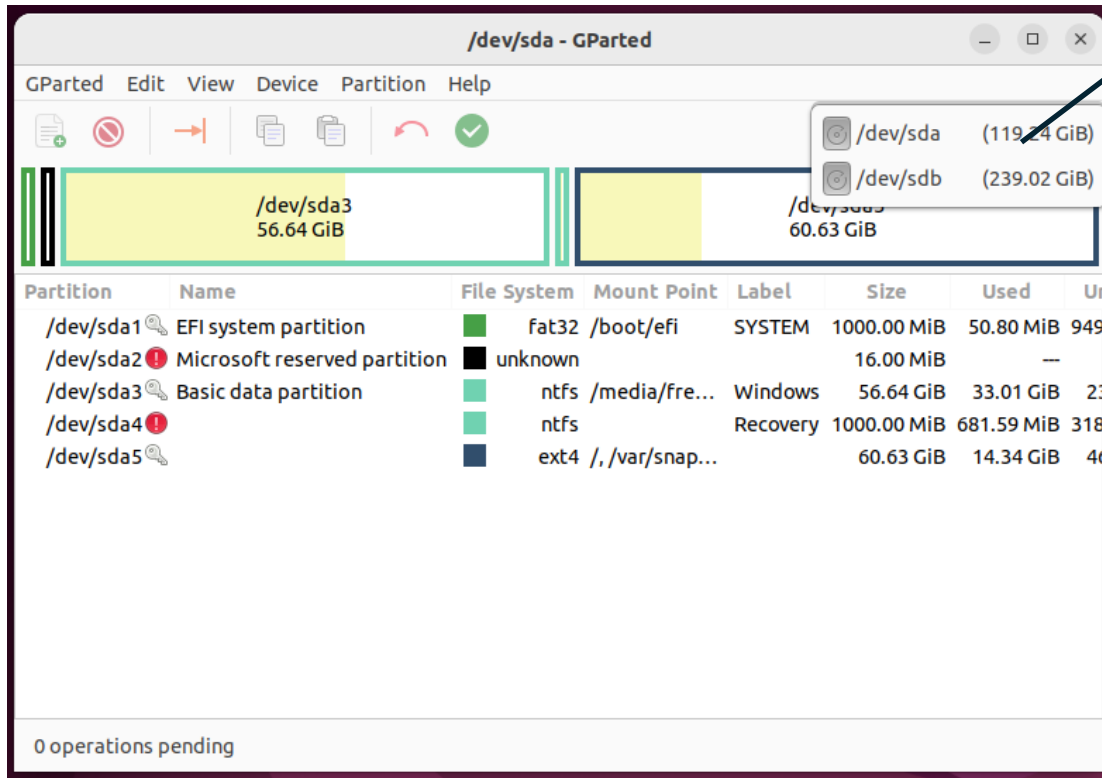
GParted will show all of the disks and the associated partitions on the system. This could be confusing as it is specific to your computer and only general information can be given.

The size of the disks will be helpful in enabling you to tell which disk is which. It is important to minimise the number of disks connected to the computer as this will reduce the confusion, so as a first step unplug any USB other than the two you need for this process. In the case of a laptop any additional SSD or hard drives are unlikely, but they may be present in a desktop or other well-equipped PC.

Take note of the drives and make sure you can identify them. In this case the disk of interest is likely to be the first one, disk0, probably sda.

Installation of a Stand-alone Bootable Linux Build on a USB (Use as a Daily Drive or Rescue Disk)

Linux does not name drives like Windows does. Instead, it sees each disk as a device like `/dev/sda` or `/dev/sdb` in a tree. Each partition is a number associated with the device. For example, the first partition on sda is `sda1` and so on. So you will see something like this



Click here to list the drives installed, in this case there are two.

`sda` is the system disk (disk0), a 120GB SSD.

`sdb` is a USB key, in this case larger than the SSD at 256GB.

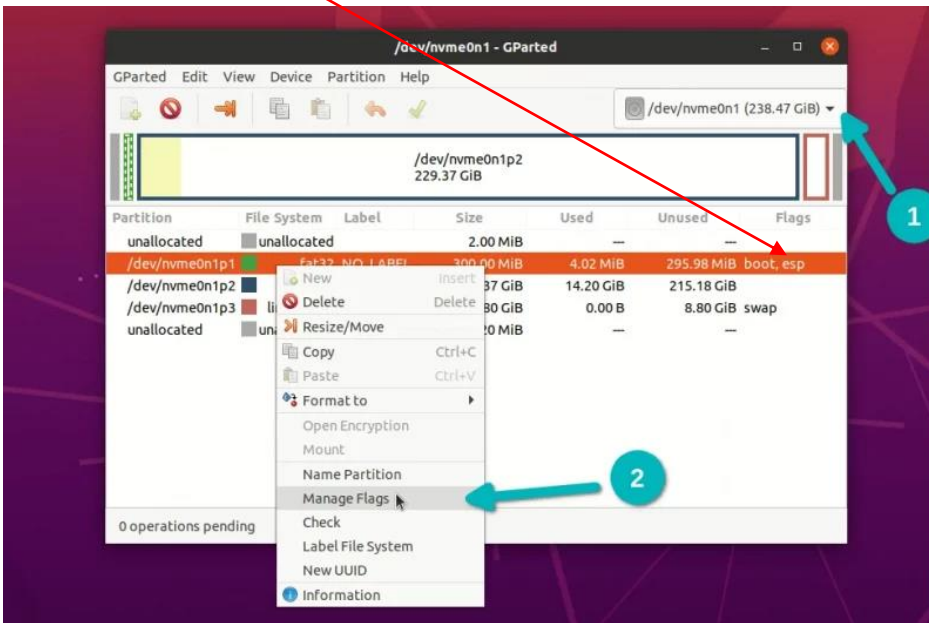
By looking at the list of drives you will be able to determine which is the system disk, which is the install USB and which is the target stand-alone USB

The system drive of a dual boot PC is shown here. Each of the partitions on disk are shown. In this case there are 5 of them numbered 1 to 5. The first (`sda1`) is the EFI system partition where the bootloaders are stored. `sda2` is the Microsoft reserved partition, leave this alone under all circumstances. `sda3` is the NTFS Windows partition. `sda4` is the NTFS Windows recovery partition. `sda5` is the ext4 Linux partition.

It is the EFI partition that is of interest.

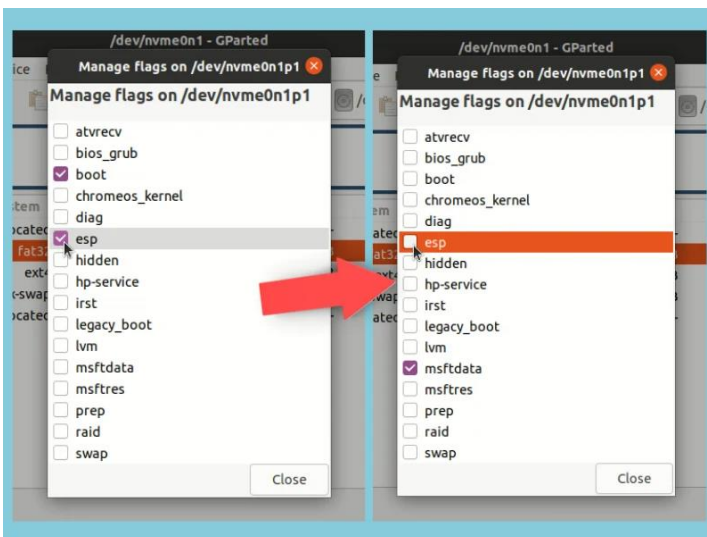
Installation of a Stand-alone Bootable Linux Build on a USB (Use as a Daily Drive or Rescue Disk)

In GParted, make sure that you have selected the system's disk. Look for the partition of around 100-1000 MB in size and flagged as esp. Right click on this partition and select 'Manage Flags':



Reset flags for the esp partition by selecting the Manage Flags option.

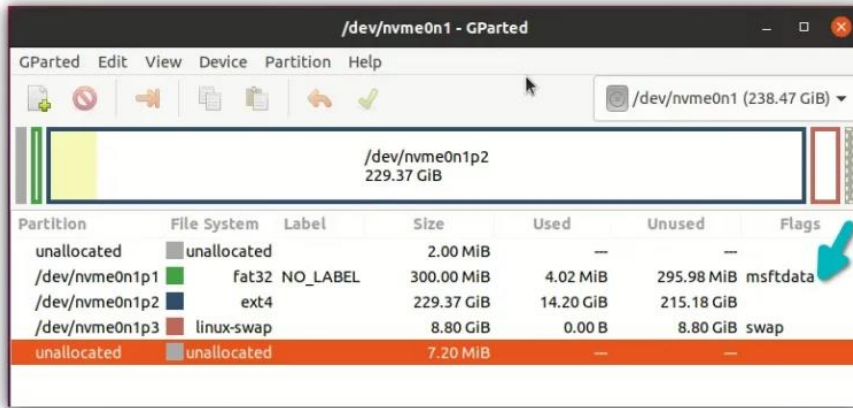
It should show esp and boot flags. If there are more flags set, take a screenshot (or a photograph) to revert to the same set of flags after the installation is complete.



Remove esp flag from the ESP partition

Deselecting the flags usually adds a msftdata flag and you can leave it like that. Changes take effect immediately.

Installation of a Stand-alone Bootable Linux Build on a USB (Use as a Daily Drive or Rescue Disk)

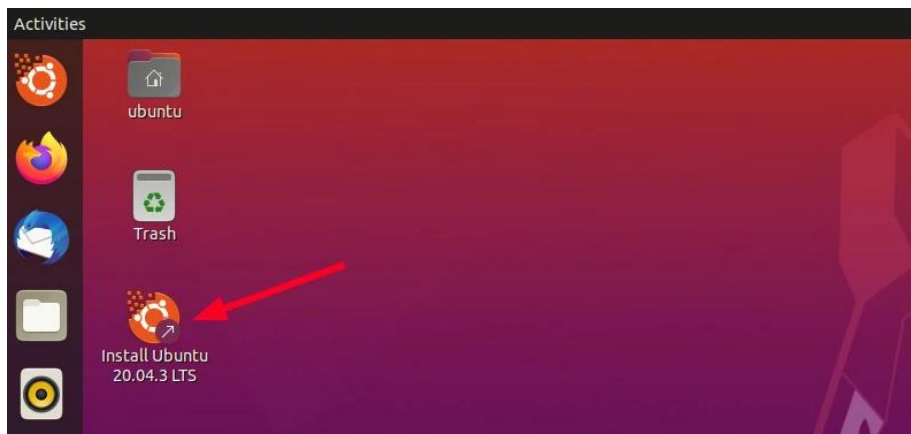


esp flags are now removed from the esp partition and it does not show as boot. **Note that in this state your computer is disabled for booting as there is nowhere to boot from that the computer can find. You must remember to reverse this step later.**

Great! Now your live system does not see the esp partition and thus the existing boot settings of the computer. Now is the time to start installing Ubuntu on the other USB.

Step 3: Installing Ubuntu on the other USB

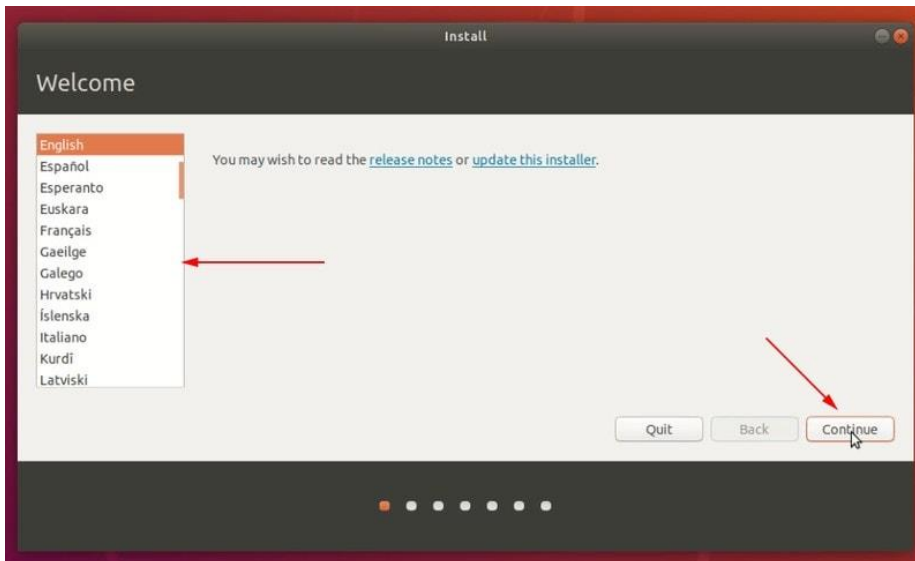
Make sure that you have plugged in the other, bigger-capacity USB as well. Double click on the install Ubuntu icon on the desktop. The icon may appear somewhere else and the window may have a different appearance.



The Ubuntu installation will start.

Installation of a Stand-alone Bootable Linux Build on a USB (Use as a Daily Drive or Rescue Disk)

You can go through the first few steps of choosing the keyboard layout and language. You are not repeating steps here, you are setting up the configuration of the install to the Linux To Go USB.



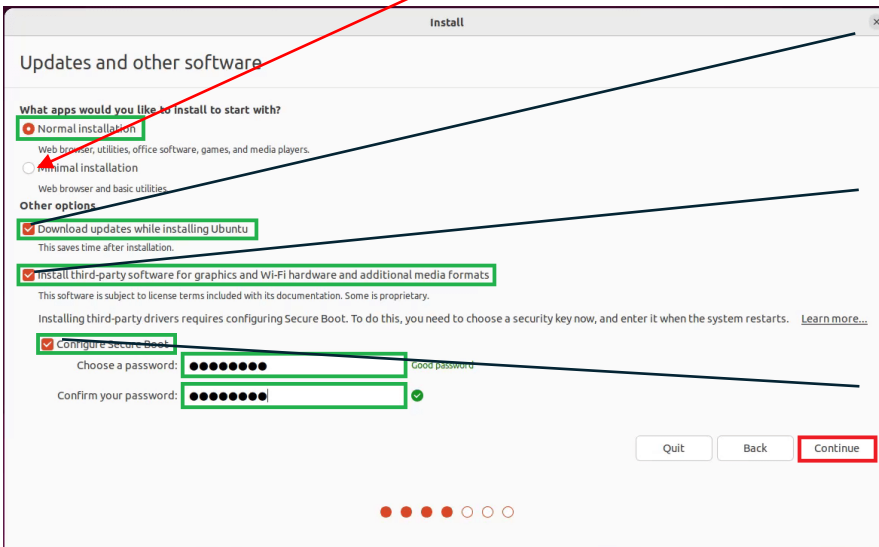
Choose your language / Choose keyboard layout

Be aware that installing on the USB, especially USB 2, will be quite a bit slower than onto an SSD. For this reason, the minimal install is a good choice, which comes with a browser and essential utilities but won't include office software, media players etc. You can install them later if you find they are needed.

Using Minimal Installation means copying less files on the USB and this will reduce the installation time.

Installation of a Stand-alone Bootable Linux Build on a USB (Use as a Daily Drive or Rescue Disk)

The next step is to choose the install type. **A minimal install is recommended.** The Linux USB To Go is not intended to be much more than a rescue tool and the full install is not required. It will also speed the install up.



Select this option to download updates during the install. Note that this requires an Internet connection

If you elected (or are forced) to install third party software check this box. **Be aware this will add steps to the install as your install is no longer signed**

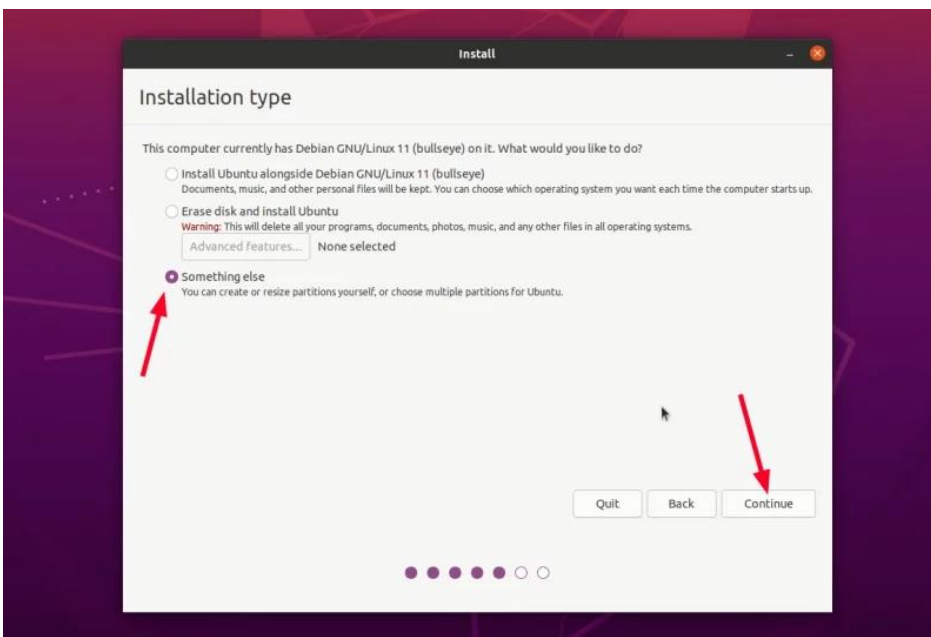
If you chose Secure Boot then enter and repeat that password. **Note that password down, you will need it later.**

If you do not want Secure Boot then untick Configure Secure Boot

The password you enter here is not the user level password you will be asked for later. This password allows you to enrol a security key called the MOK (Machine Owner's Key) as the validity of the install is lost by installing uncertified third-party software. As you go through the first boot you must certify the software, by using this password, exactly once.

Selecting minimal install will reduce the installation time. It is likely best to avoid third party software as well if you can. Some specific hardware configurations are not supported by generic drivers, and in that case third-party drivers may be the only solution. Using standard drivers will give the build the best chance of the running on a range of different hardware. Click Continue.

On the next screen, select Something Else:

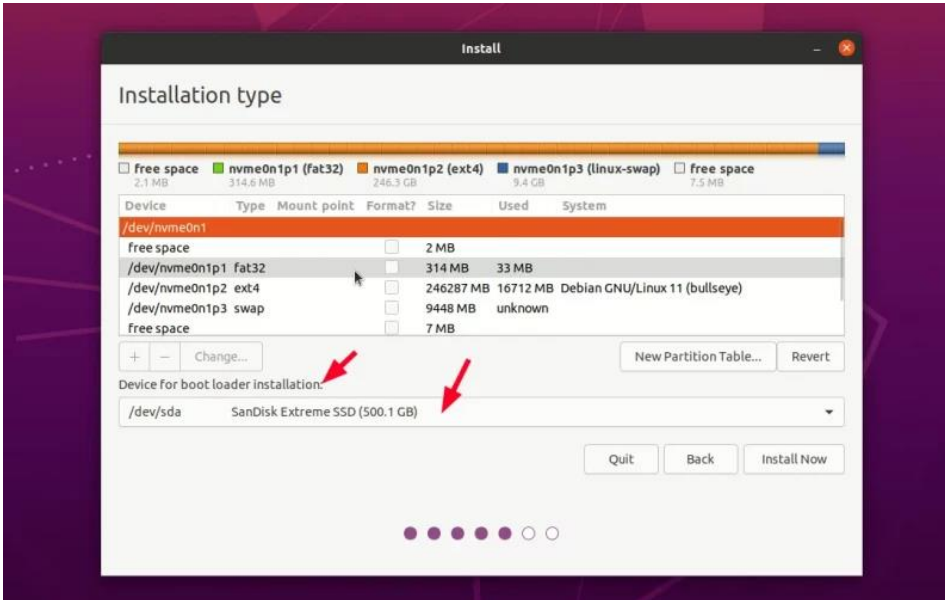


Something else option will let you access the disks and partitions, as the install wanted is not standard.

Installation of a Stand-alone Bootable Linux Build on a USB (Use as a Daily Drive or Rescue Disk)

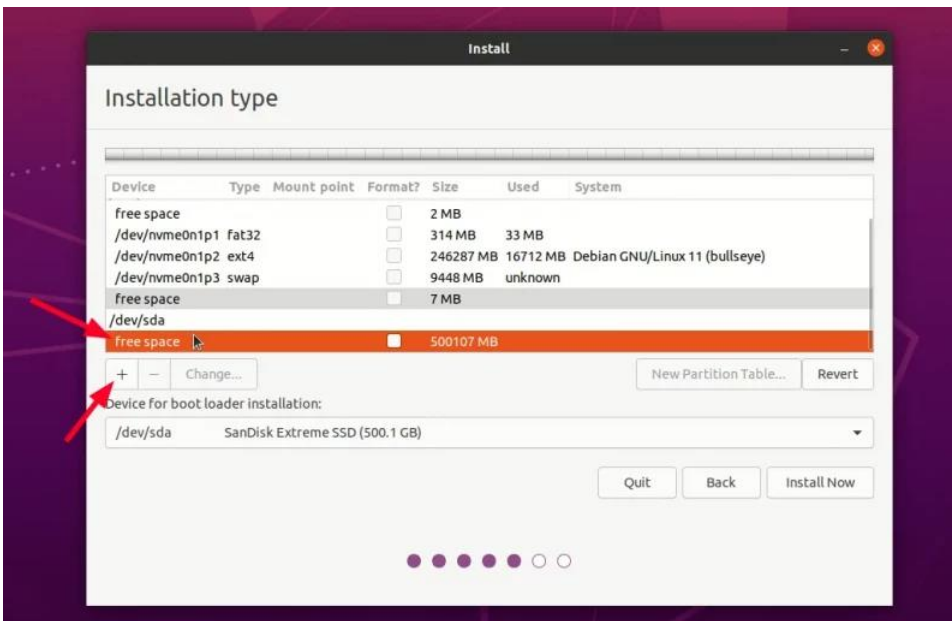
On the 'Installation type' screen, you can see all the hard disk and the USB disks listed on the main interface. You can see that the hard disk doesn't show an ESP partition because the flags have been removed.

What you should make sure here is that you have selected the correct USB (the bigger one) the Linux To Go for "Device for boot loader installation". The case shown here uses a thunderbolt SSD here which is of 500 GB capacity (bigger than the hard disk of the laptop), but your case will depend on what you have:



Location of the boot loader should be the external stand-alone boot USB.

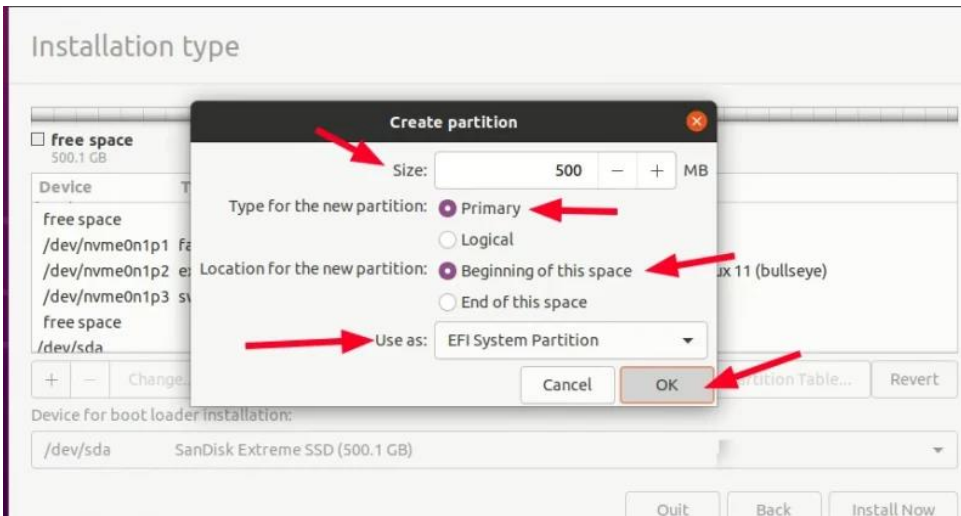
Now, select the bigger USB and delete any existing partition to make free space by clicking the - button so that the USB is blank. (Having the USB already blank speeds this up).



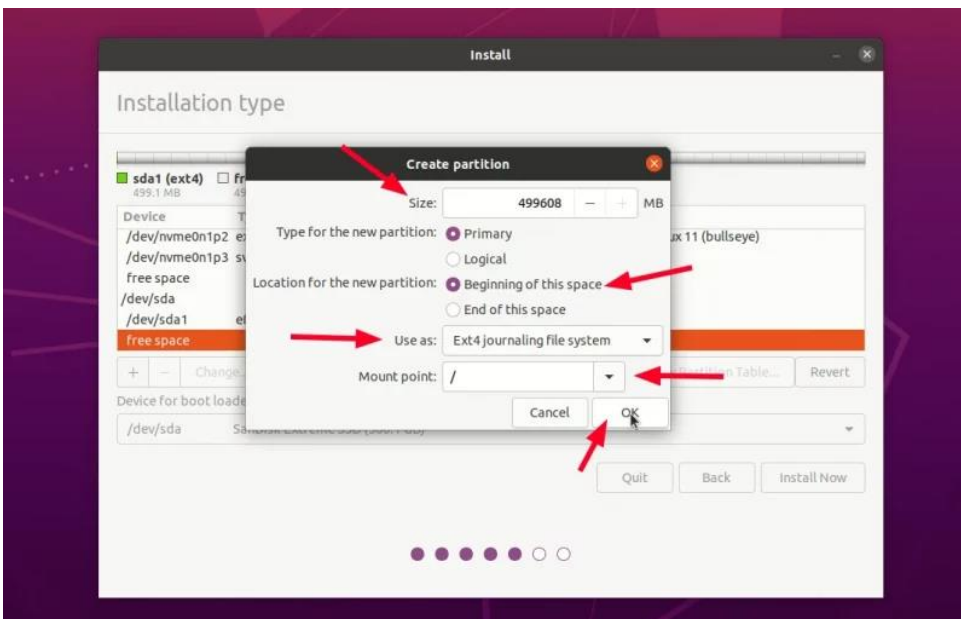
Create new partitions on the external USB where the Bootloader and the Linux OS will be installed.

Installation of a Stand-alone Bootable Linux Build on a USB (Use as a Daily Drive or Rescue Disk)

Make the first partition as ESP for the bootloader. 500MB is satisfactory for this case even though 1000MB is usually recommended. Other bootloaders in this partition are unlikely.

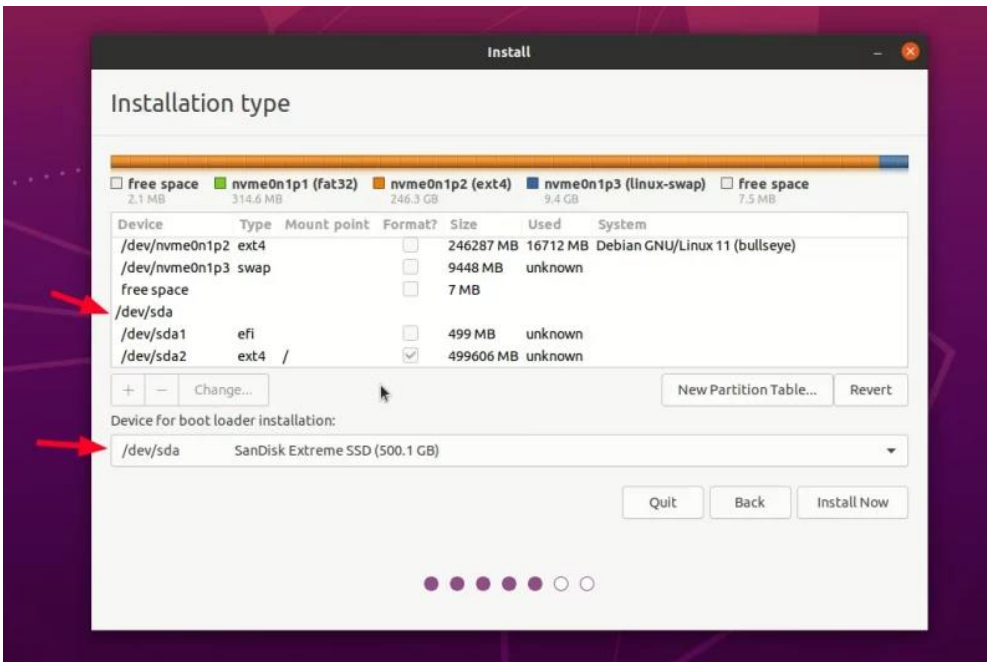


Select the remaining free space and create root with Ext4 filesystem. You could also create root, swap and home folders, but keeping it simple will work.

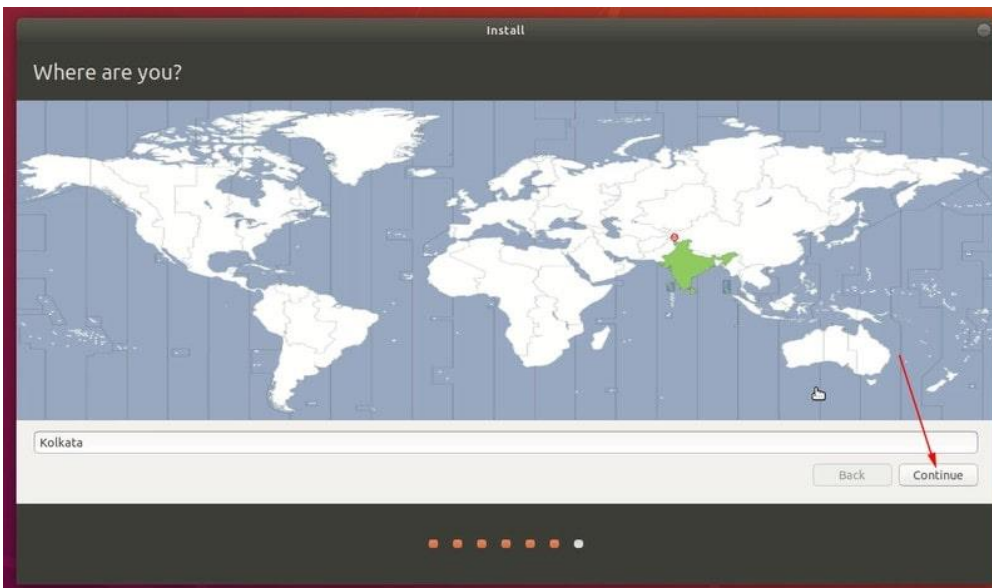


Create a root partition with ext4 filesystem where the Linux system will actually load.

Installation of a Stand-alone Bootable Linux Build on a USB (Use as a Daily Drive or Rescue Disk)



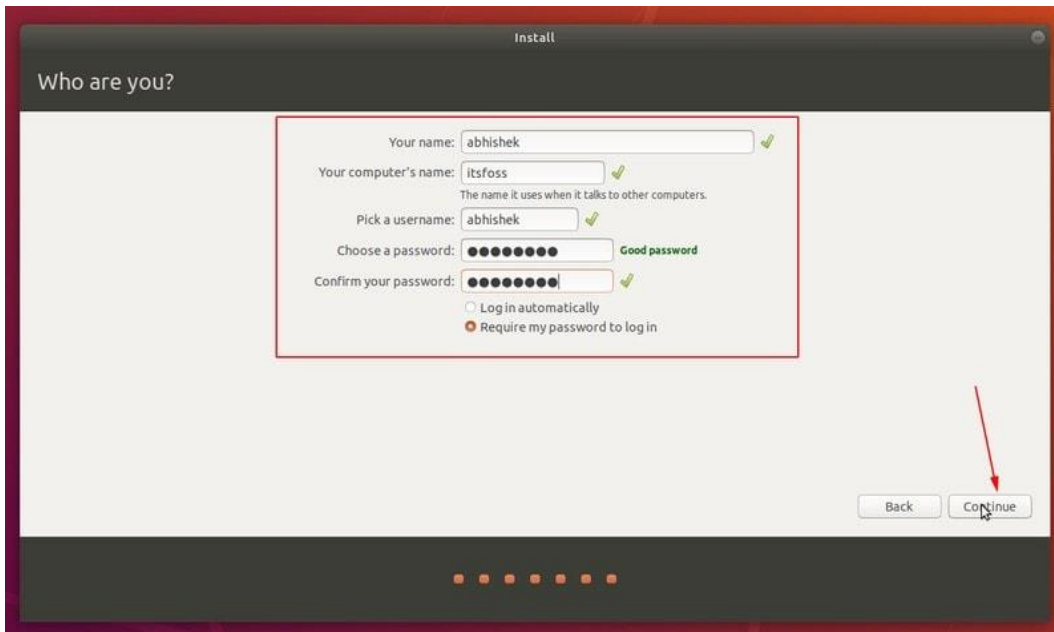
Double check the bootloader location which should be on the external Linux To Go USB drive
If it looks good then click the Install Now button and go through the usual timezone settings.



Select the timezone applicable to you.

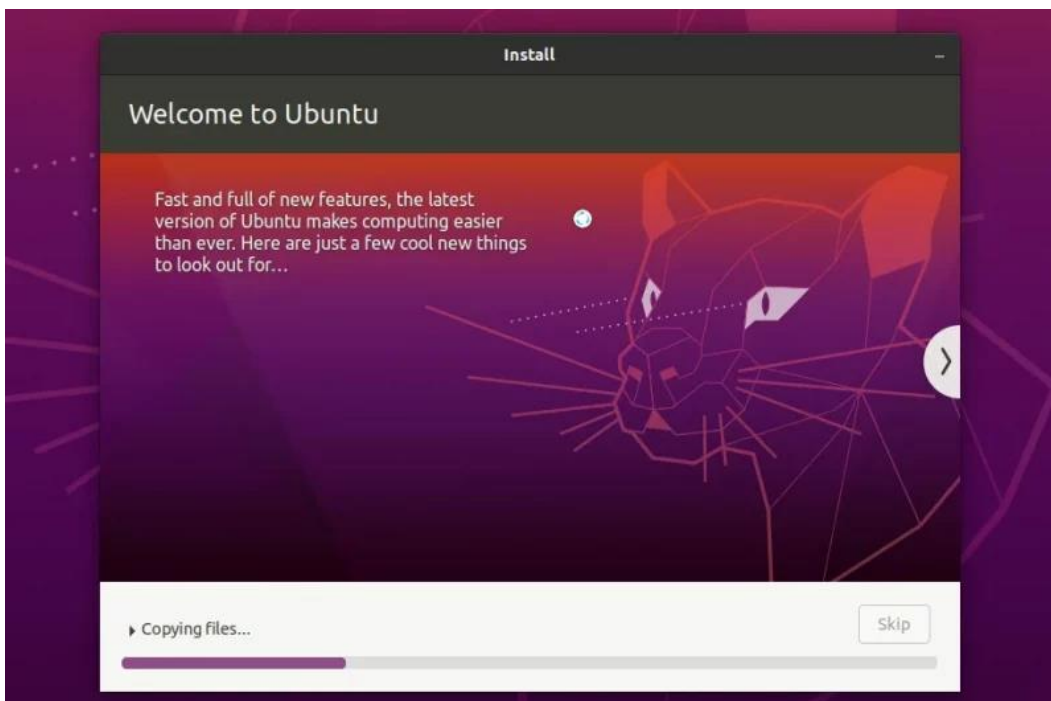
Installation of a Stand-alone Bootable Linux Build on a USB (Use as a Daily Drive or Rescue Disk)

You'll be asked to create user and password to log into the system after it has loaded. You can elect to log in without a password and for a rescue/emergency disk this could be acceptable.



Set username and password if required.

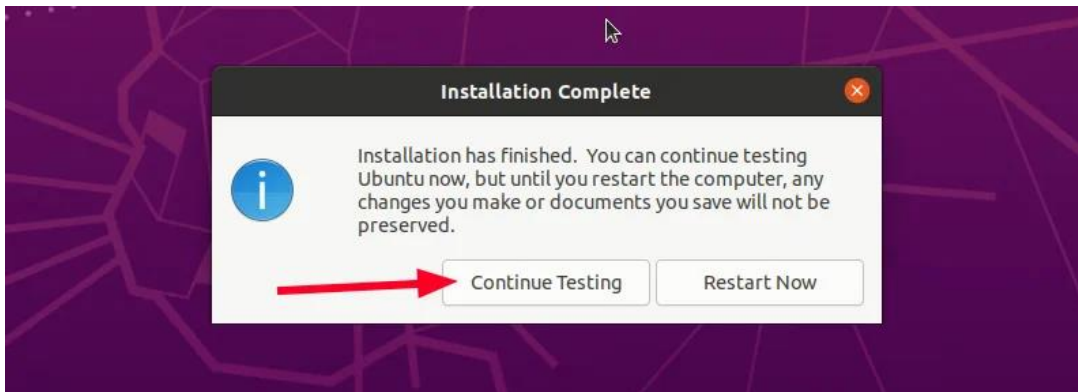
After this, it is all a matter of waiting. If you have ever installed Ubuntu, you'll surely feel that this installation is slower than usual. The screens you see may differ as different software has different screens.



Ubuntu installation is in progress. Be patient, you may leave and let the install progress if you like. For USB 2, 2 to 3 hours is not unusual.

Installation of a Stand-alone Bootable Linux Build on a USB (Use as a Daily Drive or Rescue Disk)

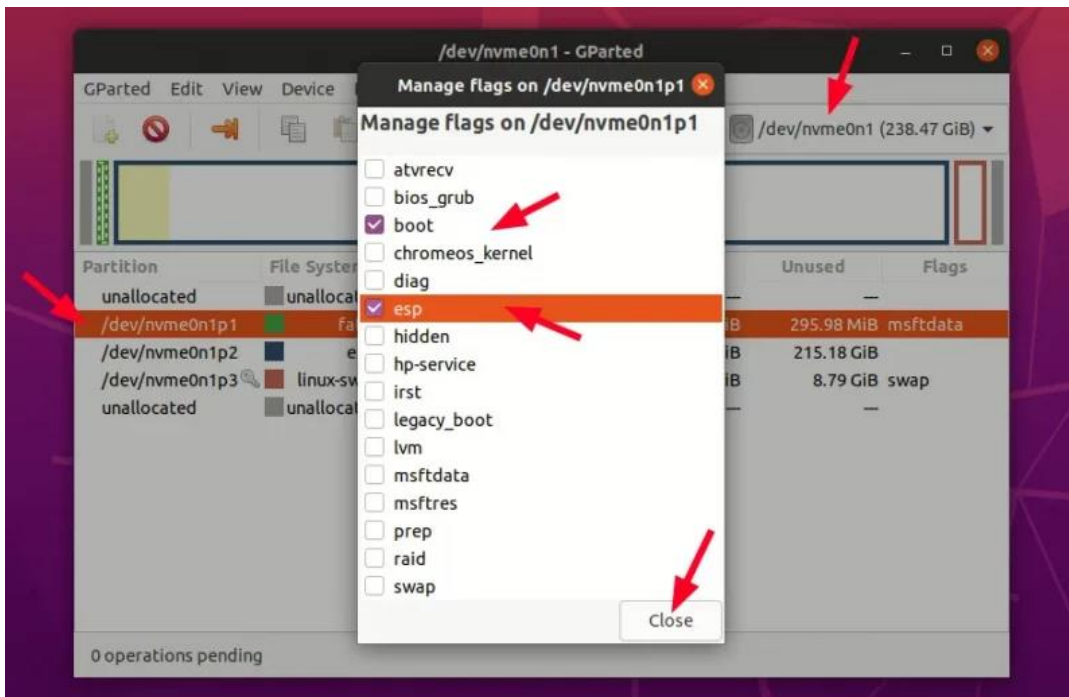
When the installation finishes, **DO NOT RESTART** right away. You have to reset the esp partition back to its original state.



Continue Testing using the live Ubuntu environment

Step 4: Re-enable ESP flags on the ESP partition

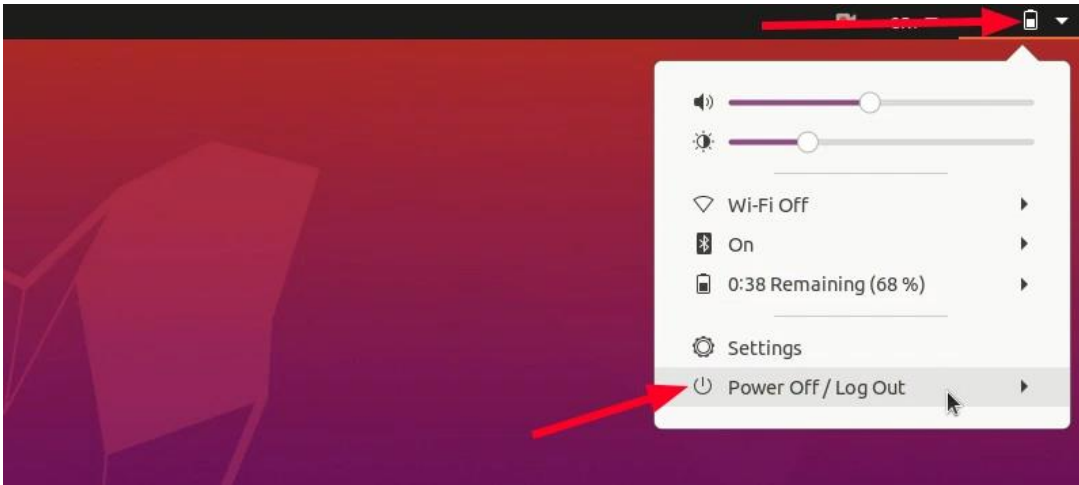
Start GParted once again. Select disk0 of your system and right click on its ESP partition which is now labelled as msftdata, right click and choose Manage Flags. Select esp flag and it should also select boot flag automatically.



Set esp flags again and click Close

Installation of a Stand-alone Bootable Linux Build on a USB (Use as a Daily Drive or Rescue Disk)

Congratulations! You have successfully installed actual Ubuntu on a flash drive. You can now turn off the live Ubuntu system.



Restart or turn off the system now.



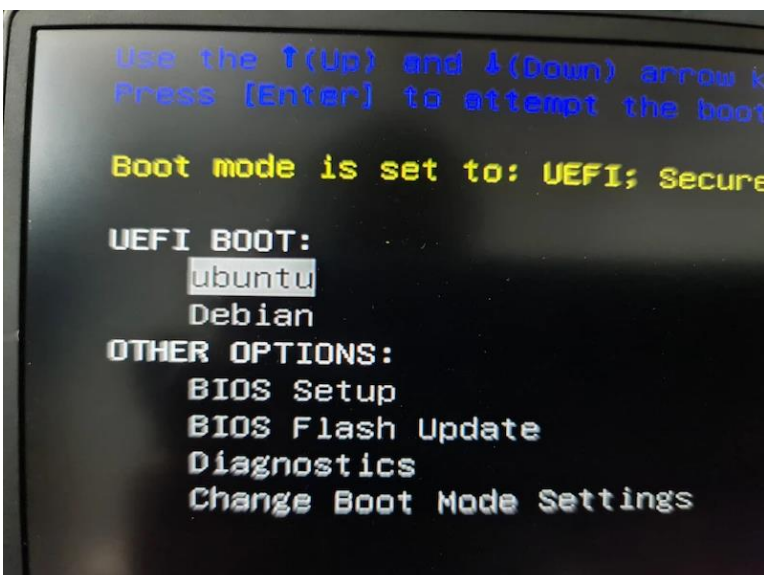
What if you accidentally hit the restart button without resetting the ESP flags?

That's bad but not too bad. Since your actual system does not have an ESP partition it won't boot without the external USB you just created. But no need to worry. You still have the live Ubuntu USB, right? Boot from it again. Start GParted and enable the correct flag on the ESP partition of the disk.

Part 3: Using the Ubuntu USB on any system

The USB you just created can be used on any computer as long as it allows to boot from the USB. In other words, the secure boot should be disabled.

The process is similar to how you booted from the install USB. You turn on the system; when the system manufacturer's logo is visible, press the keys required to access the boot settings and select the Linux To Go USB to boot from. It may be displayed with either of the two names (or more perhaps).



Your system will appear differently.
Example system shown.

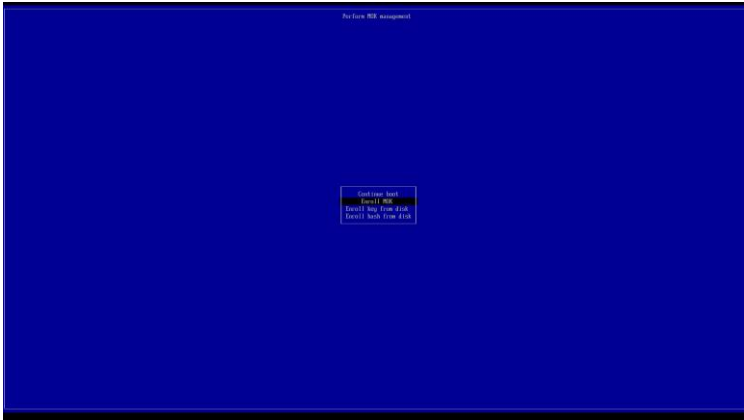
Installation of a Stand-alone Bootable Linux Build on a USB (Use as a Daily Drive or Rescue Disk)

Booting Ubuntu on the external Linux To Go USB

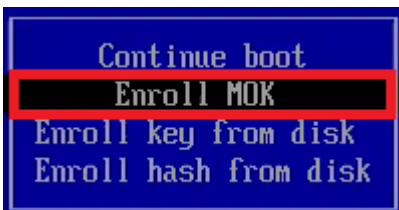
If you lost the trusted status of the build and secure boot is turned on you will need to enrol the MOK using the password you created in the build process exactly once. If you did not trigger the loss of validity you will not see these steps even if secure boot is turned on and your machine will directly boot.

Steps related to the MOK are shown in blue. You will only have to do this once!

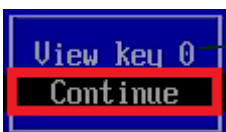
During the first time boot of the Operating System, you will be taken to the UEFI BIOS MOK Management screen:



Press the ↓ key until you get to Enroll MOK and then press ↵:

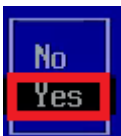


Press the ↓ key until you get to Continue and then press ↵:

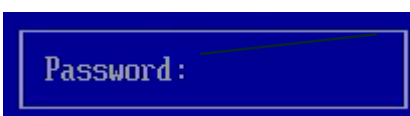


If you are curious you can view the key before you continue to enrol it

You will be asked, whether you want to enrol the keys. Press the ↓ key until you get to Yes and then press ↵:



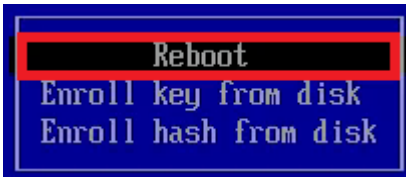
Input the Machine Owner Key (Secure Boot Password) that you created when building the Linux To Go USB and then press ↵. Note that the box does not display any characters or * as you type.



The password here is the one associated with the Secure Boot Page. It is not your login password, although you may have set them to be the same

Installation of a Stand-alone Bootable Linux Build on a USB (Use as a Daily Drive or Rescue Disk)

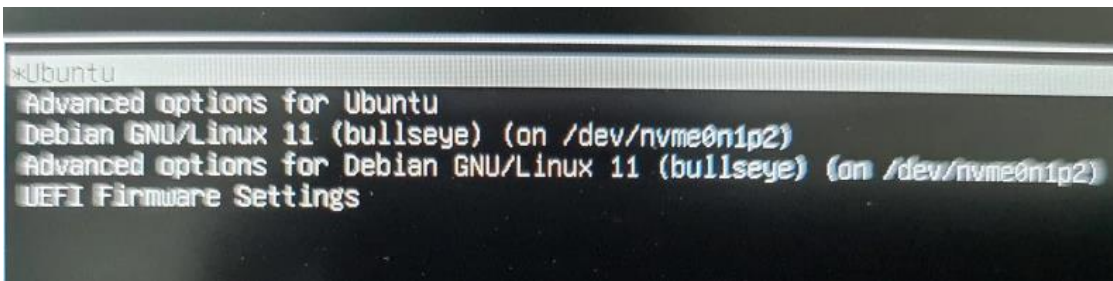
If the password is accepted, you will not see any confirmation and instead just see this screen. Reboot should be highlighted, press `↵`:



If the Machine Owner Key does not match the Secure Boot Password you will be warned and be prompted to input it again. If you have forgotten this password, go back and start at the beginning.

If successful you have enrolled the MOK and it will not trouble you further.

When you boot from the Ubuntu USB, you'll see the familiar Grub screen and you can select Ubuntu to use it:



Grub menu of Ubuntu on the Linux To Go USB. The first option will boot to the Linux To Go USB

Part 4: Getting your USBs back in normal condition

Here's the thing you have used two USBs: one for live Ubuntu and one for actual Ubuntu install.

If you want to use either of the USBs later for normal data transfer, you'll have a difficult time formatting them. When you create a live USB, it leaves the USB in a weird state and often operating systems cannot format it directly.

GParted comes to the rescue once again. It is available for Linux, Windows and macOS. Download and install it and use it to format the USB by deleting all the partitions on it and creating a new partition in NTFS or FAT32 format.

Conclusion

The neat trick of hiding the actual ESP partition from the Ubuntu installer is the key idea here. This is a better solution than removing the hard disk or having a messed up boot.