

# Compendium.

Quick-start guide  
for the Nitrux Operating System.



This is the documentation that you're looking for.

---

## Purpose of the publication.

The Compendium walks new users through the steps of obtaining a copy of Nitrux, installing it, configuring it to work with your hardware, and putting it to daily use. It aims to provide a readable general introduction and purposely gives preference to graphical tools when available. Your experience with your computer will be at its best if you're comfortable with the operating system installed on it. With that in mind, we have provided this publication to get you up and running with Nitrux.

If you are new to Nitrux or only casually acquainted with Linux based distributions, it can be challenging to understand how a Linux operating system compares with other computer systems that you may already be familiar with. Hopefully, this document will help demystify Nitrux for new-comers.

## 1.1 Getting started is easy.

To install Nitrux on your computer make sure to check the hardware requirements below. Minimum requirements indicate what we are considering to be the least powerful hardware setup to run Nitrux and still be able to use it without hurting the user experience that much. Yes, Nitrux could run in even lower-end hardware than what we list. However, the user experience would be utterly inadequate due to hardware limitations.

Therefore the recommended requirements are so that the user experience of Nitrux is optimal and the minimum requirements are so that user experience of Nitrux to be good enough.

	Minimum	Recommended
CPU	2.13 GHz Dual-Core 64-bit	2.4 GHz Quad-Core 64-bit
RAM	4GB	8GB
Storage	3.0GB	3.0GB
Display	32MB VRAM OpenGL 2.1 Support	256MB VRAM OpenGL 3.0 Support
Network	Ethernet RJ45 Port	Ethernet RJ45 Port; Wireless card, Bluetooth
Other	Motherboard with support for USB boot, EFI or UEFI	

	Minimum for VMetal	Recommended for VMetal
CPU	3.0 GHz Quad-Core 64-bit	3.0 GHz Hexa-Core 64-bit
RAM	8GB	16GB
Storage		52GB
Display	2x UEFI-compatible Graphics Processing Units (2x Discrete; 1 Discrete/1 Integrated; 1 Integrated/1 External)	
Network		Ethernet RJ45 Port; Wireless card, Bluetooth
Other		Motherboard w/ support for USB boot,UEFI, and IOMMU; CPU support for Intel VT-d and VT-x or AMD-Vi and AMD-V

# 1.2 Obtaining and flashing an ISO

Nitrx is distributed as disk image file in the ISO 9660 file system format.

## 1.2.1 Downloading a Nitrx ISO file.

The Nitrx ISO file can be downloaded in two ways. The first is directly from our website; this ISO is a traditional Linux distribution based on Ubuntu. It uses the Calamares installer and includes NX Desktop..

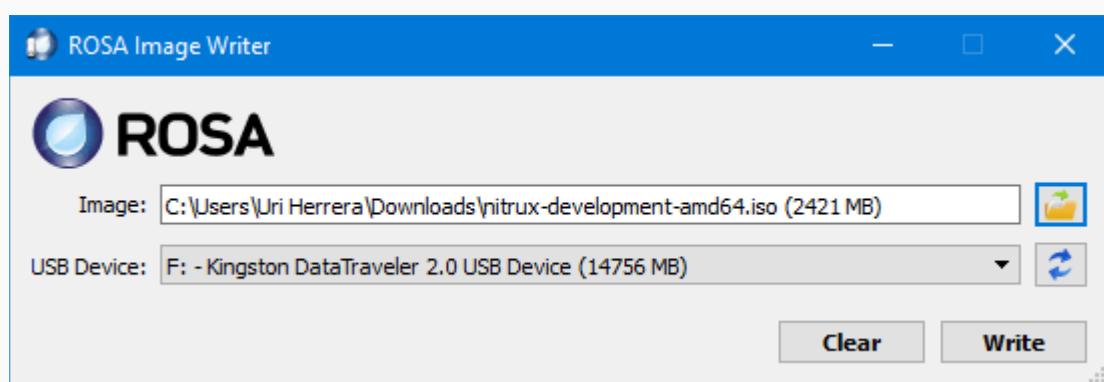
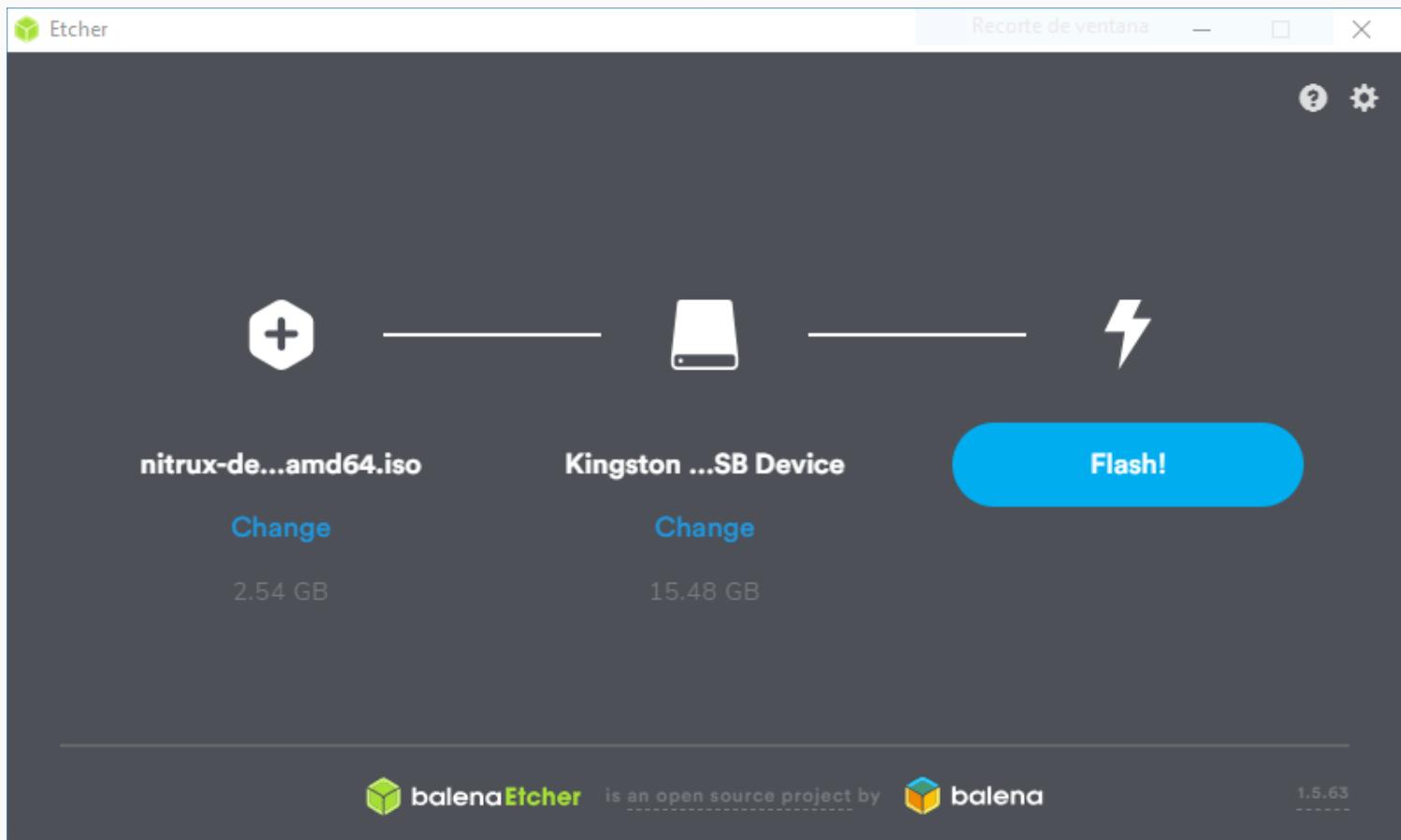


# 1.2 Obtaining and flashing an ISO

To flash our ISO to a USB, we recommend that you use balenaEtcher, Rufus or ROSA Image Writer if you're using Windows. If you're using Linux, you can use any application such as the command-line program 'dd,' ROSA Image Writer, KDE ISO Image Writer, balenaEtcher, etc.

## 1.2.2 Flashing the Nitrx ISO file.

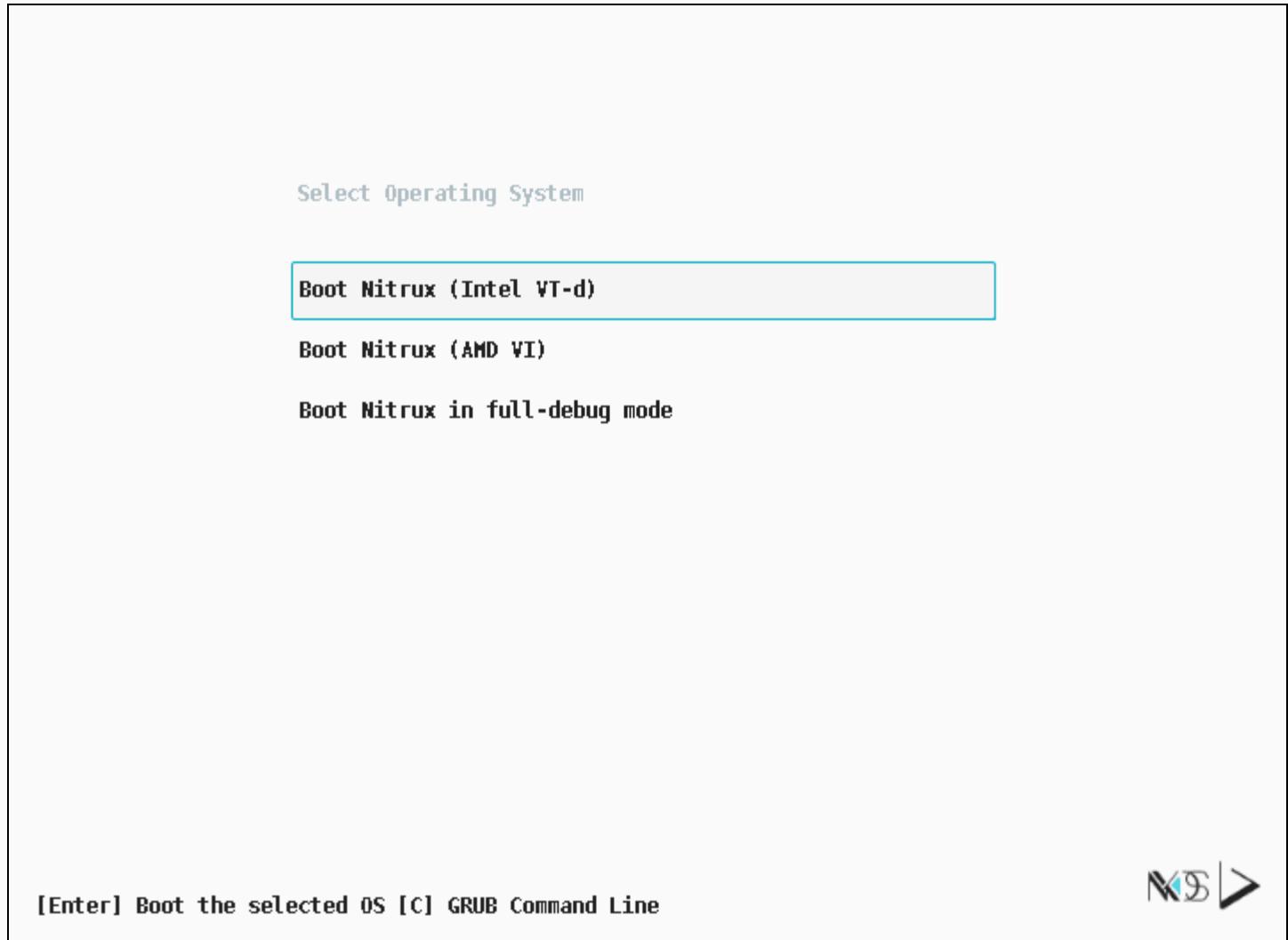
You can flash our ISO using one of the programs shown below.



### 1.3 Booting Nitrux

When booting up Nitrux, you will be greeted by two screens. The first one will be the znx boot menu where you will be able to select the deployed operating system.

This is the Nitrux boot menu. The two entries refer to the CPU technologies that Nitrux makes use of to make VMetal work. Each entry will have sub-entries that refer to the GPU in the computer.



## 1.4 Deploying and Updating Nitrx with ZNX

### 1.4.1 Installing vs. Deploying

Znx is an operating system manager. It manages the lifetime of OSes that are deployed with it. Znx is not an installer, a container, a program to flash USBs (it's not a replacement for 'dd' or anything similar), or virtualization software.

**What znx does is frugal installations.** *"A frugal installation only occupies one folder in a partition, and the rest of the partition can be used for anything else. Other Linux distributions, for example."* Meanwhile, traditional Linux distributions do a full installation, "A full installation is where Linux occupies an entire partition, and in that partition you will see the folders /bin, /sbin, /opt, /etc/, /sys, /proc, /tmp, /dev, /usr, /run, /lib, and more."

An installer such as Ubiquity, Calamares (KPM Core), Anaconda, and every other installer works the same, they extract the contents of the SquashFS file inside the ISO and place the contents on a partition of the storage device. That is why we don't refer to znx "installing" an OS; instead, we use the word **deploy**. Because znx isn't extracting the SquashFS file from the ISO, it's booting the ISO directly, and data is preserved on the storage device using OverlayFS.

First identify the device where you want deploy Nitrx, then initialize it. You use the commands 'blkid' or 'fdisk' to list your attached storage devices. Be warned that initializing the storage device will wipe the device clean.

Then, proceed to deploy Nitrx to the **device**. Please note that the initialization and the deployment are done to the storage **device**, e.g., **/dev/sda** and not a **partition**, e.g., **/dev/sda1**, znx will automatically handle partitioning. Deploying Nitrx will convert the target storage device from Master Boot Record or MBR to GUID Partition Table or GPT. If you're using an NVME device use **/dev/nvmeXnY** where X is the port and Y is the device number, as mentioned before, do not enter a partition.

Replace **\$version** with any name, don't use spaces or hyphens, and point to the path of the ISO file, this can be a local file or a remote URL.

Finally, make sure that the ISO has been deployed to the device by emptying the write cache buffers.

```
sudo znx init /dev/sdX
sudo znx deploy /dev/sdX nitrx/$version nitrx.iso
sync
exit
```

# 1.4 Deploying and Updating Nitrx with ZNX

## 1.4.2 Updating

Nitrx provides updates to the operating system every month. These include security updates, bug fixes or newer versions of the software. All deployed systems should be updated regularly to ensure the system operates most efficiently and securely possible.

To get verbose messages during system updates use Station, the Maui Terminal emulator. Updating is easy, enter the commands to download and install the available updates for Nitrx. You can also update Nitrx from a different Linux system if you have the Appliance of znx in your \$PATH.

```
sudo znx update /dev/sdX nitrx/$version
exit
```

# 1.5 Adding New Software

## 1.5.1 AppImages

You can add new applications by downloading AppImage files.

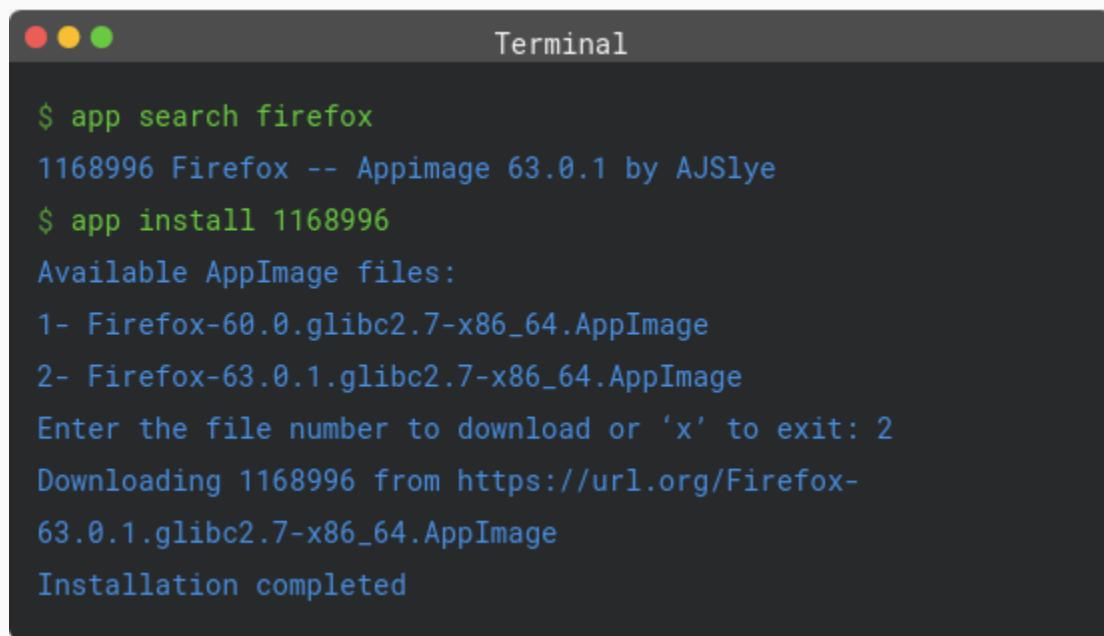
AppImages come with all dependencies that cannot be assumed to be part of each target system in a recent enough version and will run on most Linux distributions without further modifications.

Some of the significant software is already available as an AppImage, GIMP, Inkscape, LibreOffice, ONLYOFFICE, Blender, Krita, among others. The use of an AppImage by a developer to distribute their software allows them to target Linux as a whole instead of targeting a specific distribution.

We emphasize the use of AppImages in Nitrx as it is a more straightforward way of managing end-user applications.

You can download new AppImages from the web from sites like <https://appimage.github.io> or <https://www.appimagehub.com/>. Moving AppImage files to ~/Applications will add a launcher to the application menu, and it has the additional advantage of using firejail, which will serve as a sandbox for AppImage files.

You can also use the Terminal to obtain new software by using the command “app” (without quotes). The AppImages that are downloaded using this command are retrieved from AppImagehub.com.



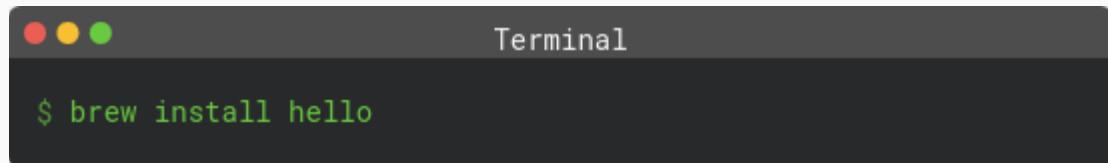
The screenshot shows a terminal window titled "Terminal". The user has run the command "\$ app search firefox" and received the output "1168996 Firefox -- Appimage 63.0.1 by AJSlye". They then ran "\$ app install 1168996" and were prompted to choose an AppImage file. They selected option 2, "Firefox-63.0.1.glibc2.7-x86\_64.AppImage". After entering the file number, they were asked to enter the file number again to confirm. They then saw the message "Downloading 1168996 from https://url.org/Firefox-63.0.1.glibc2.7-x86\_64.AppImage" and finally "Installation completed".

```
$ app search firefox
1168996 Firefox -- Appimage 63.0.1 by AJSlye
$ app install 1168996
Available AppImage files:
1- Firefox-60.0.glibc2.7-x86_64.AppImage
2- Firefox-63.0.1.glibc2.7-x86_64.AppImage
Enter the file number to download or 'x' to exit: 2
Downloading 1168996 from https://url.org/Firefox-
63.0.1.glibc2.7-x86_64.AppImage
Installation completed
```

## 1.5 Adding New Software

### 1.5.2 Homebrew

In addition to ApplImages Nitrux supports installing software using [Homebrew](#). Homebrew, the MacOS package manager allows users to install software to their home directory.



# 1.6 Wondering what makes Nitrx different?

### System

Nitrx is an operating system based on [Linux](#). Nitrx can be deployed without any need for a traditional installation. The operating system as a whole resides in a single file and directory on the partition `/ZNX_DATA/` of your device, making it easier to organize with your other data.

### Hardware Drivers

Most hardware drivers are included out-of-the-box. You don't have to hunt down manufacturer-provided drivers for every bit of hardware on your Linux system and install them. Your Linux system should automatically detect your device and use the appropriate hardware drivers.

### System Tools

**Znx** is an operating system software manager and update program that operates at a file-level to enable verifiable integrity and update efficiency.

### System Management

Nitrx promotes regular and automated updating of software to ensure the integration of new enhancements and security fixes. Because **znx** operates at the individual file-level instead of a package-level, Nitrx updates are small and fast. On many Linux distributions, upgrades to a particular software package require the whole software package to be downloaded and replaced – even for one line of code. In Nitrx, updates are generated using the [zsync](#) tool. Zsync calculates the difference between two Nitrx versions and makes available *binary deltas*, which contain only the changed portion of files. This *binary delta technology* means **znx** on Nitrx systems needs to download and apply a small fraction of a package to receive an update.

**Znx** operates against a published manifest of files for a particular Nitrx version that contains the unique hash of each file; this is the basis of the **znx update** subcommand, which allows a Nitrx system to check for and remediate any discrepancies to system files. As necessary, **znx revert** provides a useful way for software developers to return to a known filesystem state.

### Software

Nitrx approaches software management differently than many other Linux-based operating systems. Instead of deploying granular software packages, Nitrx uses the concept of ApplImages. Each ApplImage encapsulates a particular use-case, which is enabled by composing all the required upstream open-source projects and packages into one logical unit.

This ApplImage-based approach offers some unique advantages:

- ApplImages include all associated runtime dependencies.
- Software package dependencies are solved, and file-level conflicts do not occur on the target system after an update.
- All combinations of ApplImage versions can co-exist on a Nitrx.

# 1.6 Wondering what makes Nitrux different?

### Software Management

Linux-based operating systems contain the code of several hundred, if not thousands, of open source projects. To make this manageable, distributions use a concept called “packages” to configure and compile the source code of these projects into binaries.

Many distributions then split the content of these compiled packages into so-called sub-packages, which are the granularity at which these distributions deploy their software. With those kinds of distributions, system administrators can then install and update sub-packages individually or as a set, using tools such as “yum” and “apt-get.”

Nitrux takes a slightly different approach. While we also use the concept of packages to manage to compile source code into binaries, we do not use the package concept to deploy software. Instead, we make use of ApplImages. Each ApplImage contains as many or as few open source projects needed to provide a functional program.

To make it easier for users to manage their ApplImages we have added the console-based [ApplImage CLI Tool](#). This tool is available as the command **app** and can be used in the following way.

### Data Persistence

With znx, there are only two partitions on a storage device, the ESP and the Data partition. The Data partition is where znx saves the ISO files; it too will contain the data for every folder that utilizes the OverlayFS. The Data partition occupies the rest of the available space in the storage device.

OverlayFS provides a great way to merge directories or filesystems such that one of the filesystems (called the “lower” one) never gets written to, but all changes are made to the “upper” one. Brought into the Linux kernel mainline with version 3.18, OverlayFS allows you to overlay the contents (both files and directories) of one directory onto another.

We store data directly on the device, without intermediate files.