Installing an OS in a VM

Galij Sunuwar

July 2025

Contents

1 Objective

1. To install an OS in a virtual machine manager
2. To document the process of its installation

2 Introduction

2.1 Virtualization

Virtualization is the process of creating virtual computers on a single physical machine. Virtualization is the process of dividing the physical hardware of a computer into many logical parts.
It lets you run several virtual computers on one real computer so that you can use its full power and do multiple tasks at once. Using software such as Hypervisor, virtualization creates an abstraction layer over computer hardware, dividing the components of a single system into multiple virtual machines.
Virtualization is the process of creating a virtual representation of hardware such as server, storage, network, or other physical machines. It supports multiple copies of virtual machines(VMs) to run on one physical machine each with their own operating system and programs.

2.2 Virtual Machine(VM)

After installing virtualization software on your computer, you can set up one or more virtual machines. The Virtual Machines are similar to other applications on our computer. The actual computer is known as 'Host', whereas the virtual computers are known as 'Guests'.
Each guest can have its own operating system. Every guest behaves as a normal computer. It has its own configurations, programs, and settings. The resources, such as the processor, memory, and storage, are all accessed by the virtual machines, but they appear and function exactly like a real computer.
Virtualization is made possible with a hypervisor, also called a virtual machine monitor (VMM). This lightweight software layer manages virtual machines as they run alongside each other.

2.3 Virtualization Software

Virtualization software allows users to run multiple operating systems or applications on a single physical machine, optimize hardware use, and simplify management.
These software make it possible for one to achieve virtualization by providing the platform to create virtual machines. One can choose the desired VM specifications, the operating system to be installed, etc.
Popular options include Oracle VirtualBox, QEMU, VMware Fusion, and Proxmox VE, each offering unique features for different needs.

Oracle VirtualBox

VirtualBox is a general-purpose full virtualization software for x86_64 hardware (with version 7.1 additionally for macOS/Arm), targeted at laptop, desktop, server and embedded use.
Some of the main features of Oracle VirtualBox are:

Portability: It is a so-called hosted hypervisor, sometimes referred to as a type 2 hypervisor meaning it requires an existing OS to be installed.
It can be installed on Linux, Windows or MacOS. It being multi-platform, it is functionally identical on all of the host platforms, and the same file and image formats are used. This enables one to run virtual machines created on one host to be used on another.
Also, VMs can be easily imported and exported using the Open Virtualization Format which is an industry standard so you can import the OVFs created in a different virtualiztion software too.

Guest Addtions: shared folders, seamless windows, 3D virtualizations: The Oracle VirtualBox Guest Additions are software packages that can be installed inside of supported systems to improve their performance and to provide additional and communication with the host system.
After the Guest Additions, a virtual machine will support adjustment of video resolutions, seamless windows, accelerated 3D graphics and more.
In particular, Guest Additions provide for shared folders, which let you access files on the host system from within a guest machine.

Multigeneration branched snapshots: Oracle VirtualBox can save arbitrary snapshots of the state of the virtual machine. You can go back in time and revert the virtual machine to any such snapshot and start an alternative VM configuration from there, effectively creating a whole snapshot tree.

Comprehensive Hardware Support

VM groups

Clean architecture and unprecedented modularity

Remote machine display

QEMU

The **Quick Emulator (QEMU)** is a free and open-source emulator that uses dynamic binary translation to emulate a computer's processor; that is, it translates the emulated binary codes to an equivalent binary format which is executed by the machine. It provides a variety of hardware and device models for the virtual machine, enabling it to run different guest operating systems.. QEMU can be used with a Kernel-based Virtual Machine (KVM) to emulate hardware at near-native speeds. Additionally, it supports user-level processes, allowing applications compiled for one processor architecture to run on another.
QEMU supports the emulation of x86, ARM, PowerPC, RISC-V, and other architectures. QEMU is free software developed by Fabrice Bellard. Different componenets of QEMU are licensed under the GNU General Public License (GPL), BSD license, GNU Lesser General Public License (LGPL), or other GPL-compatible licenses.
It has mutliple operating modes:

- User-mode Emulation
- System Emulation
- Hypervisor Support

Supported disk image formats:

- Linux cloop
- Bochs
- macOS Universal Disk Image Format (.dmg)
- VirtualBox Virtual Disk Image (.vdi)
- Virtual PC Virtual Hard Disk (.vhd)
- Virtual VFAT
- VMware Virtual Mahcine Diks (.vmdk)
- Raw images (.img)
- CD/DVD images (.iso)

It has the following features:

- Full System Emulation
- User-Mode Emulation
- Hardware Virtualization
- Snapshotting
- Live Migration
- Versatile Disk Image Formats

2.4 Use Cases

1. OS Testing
2. Software Development and Testing
3. Server and Network Manipulation
4. Safe environment for malware analysis

3 Tools/ Requirements

Manjaro is a free and open-source linux distribution based on the Arch Linux operating system with a focus on user-friendliness, accessibility, and imporoved software testing and stability compared to its upstream sources. It uses a rolling release update model with Pacman-derived package managers.
I chose this distribution as it was easy to set up and use and it was suitable for both beginners and experienced users. It was also pretty easy to install.
I used **Oracle's Virtualbox** as the virtualization software to create a virtual machine for installing manjaro. The flavour of manjaro I chose was the **XFCE** version which is known to be lightweight compared to its other versions.

4 Procedure

4.1 Installing Virtualbox

There are various programs that offer virtualization and Orcale's Virtualbox Manager has a beginner-friendly UI so I decided on Virtualbox.

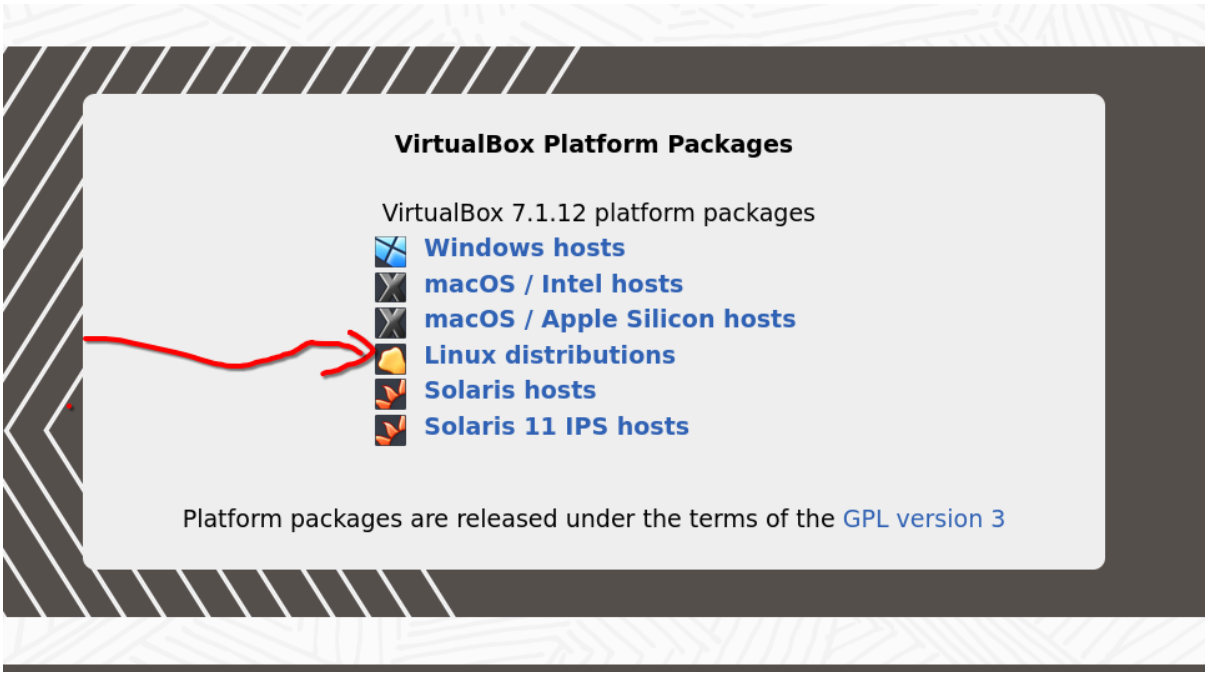

Figure 1: Oracle VirtualBox Webpage

4.2 Selecting the OS

I decided to install Manjaro OS based on Arch Linux, which uses the Pacman package manager, running XFCE as the desktop environment as it is lightweight to its KDE counterpart.

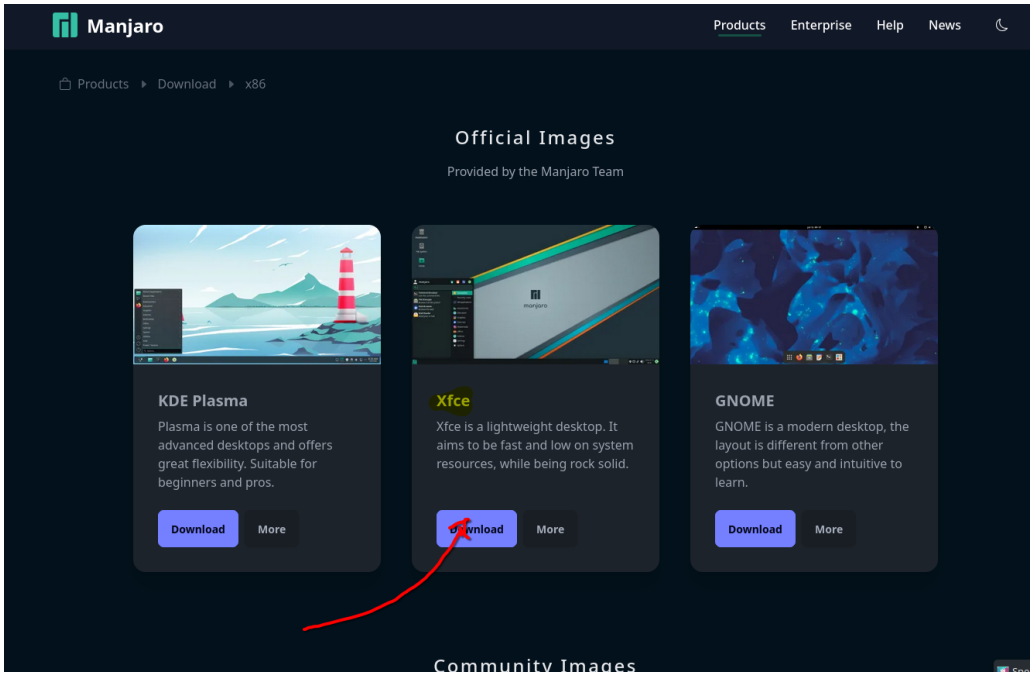

Figure 2: Official Manjaro Webpage

4.3 Creating the VM

Oracle VirtualBox provides a beginner-friendly interface along with complex features to run OS on.

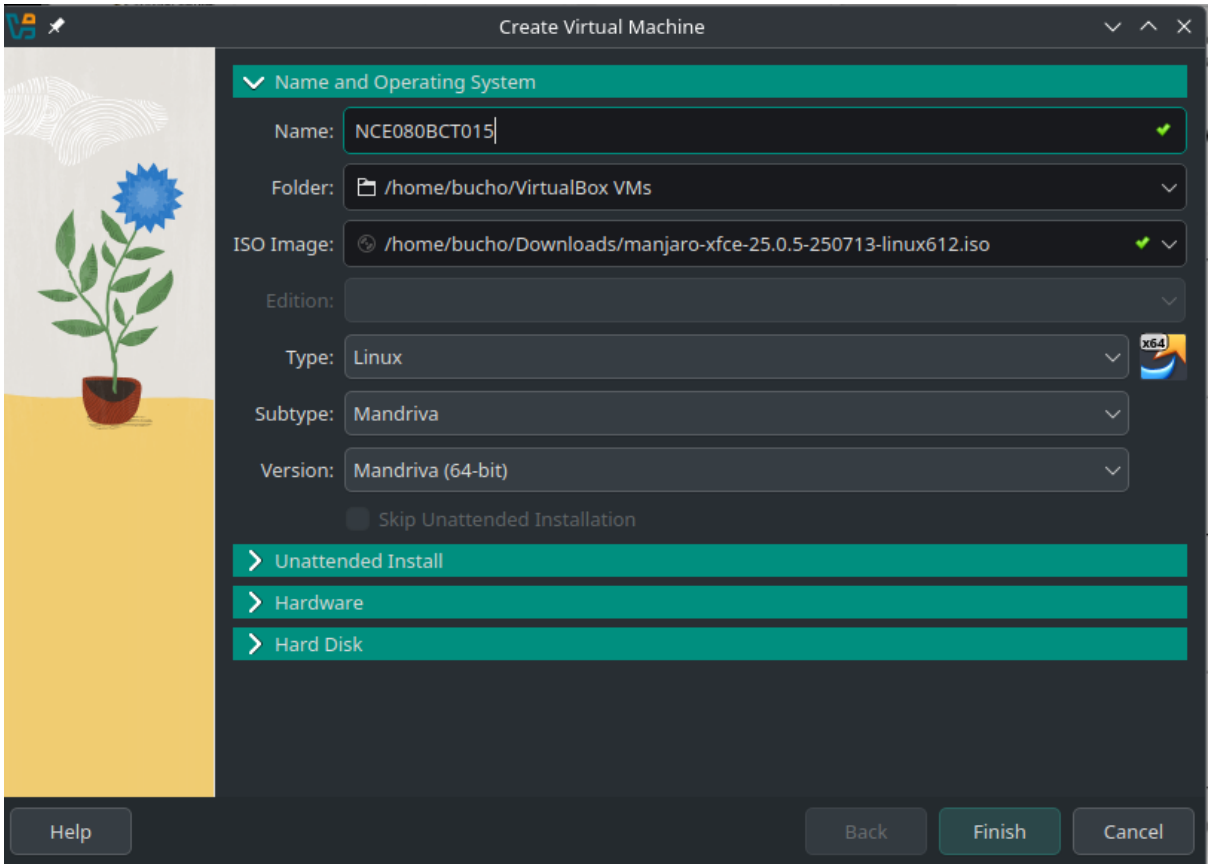

Figure 3: Naming the VM

I began by naming my VM as well as setting the ISO image I downloaded as shown in Figure 3.
Then, came the step of allocating the required memory, processors as well as the storage to the VM. Note: No more than the host's specifications could be provided to the guest.

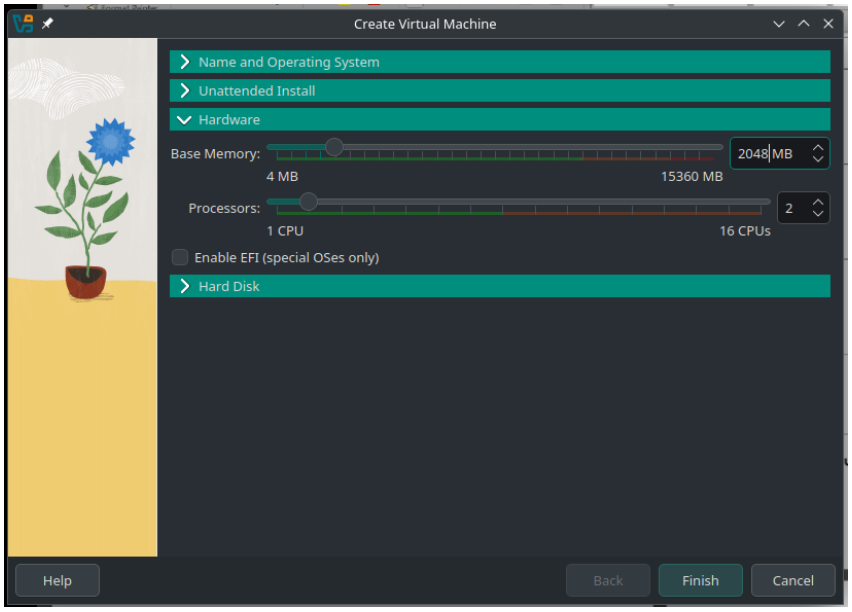

Figure 4: Memory and Processor allocation

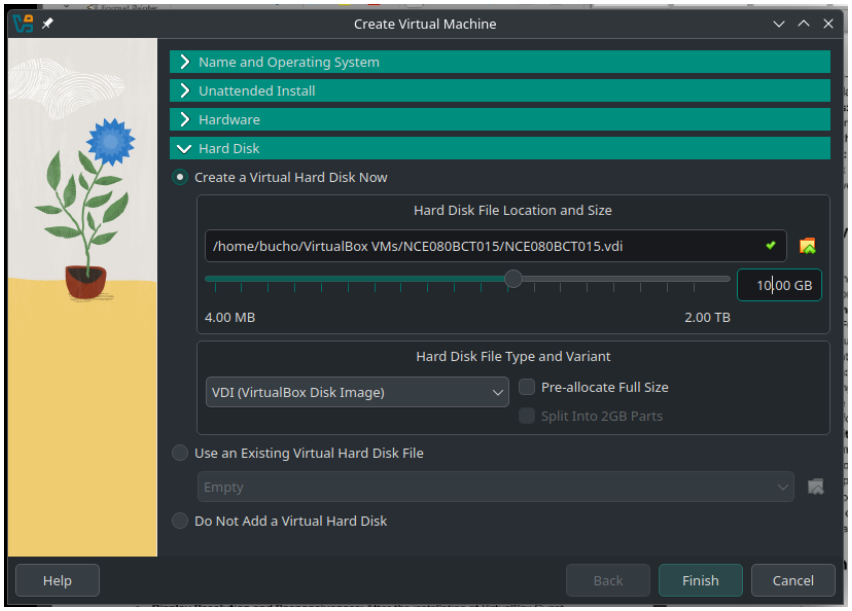

Figure 5: Storage allocation

And finally, I setup the video memory and configured the network of the VM.

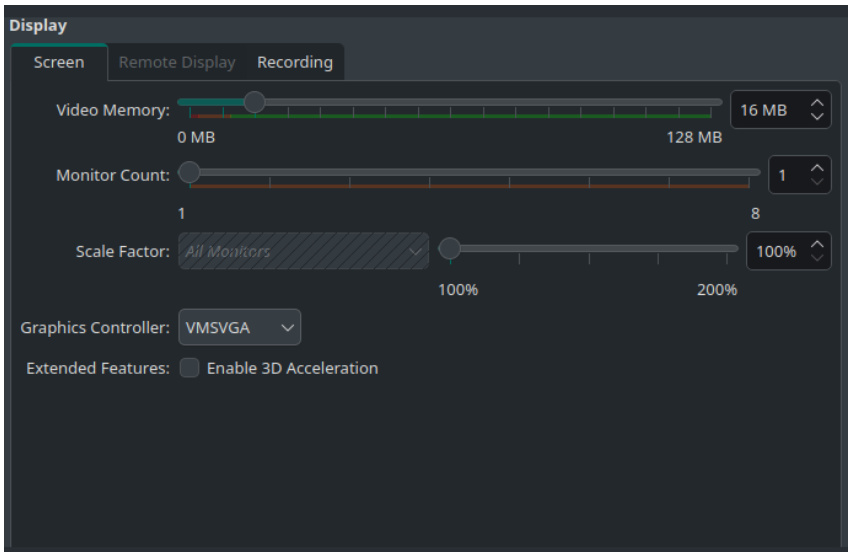

Figure 6: Video Memory Allocation

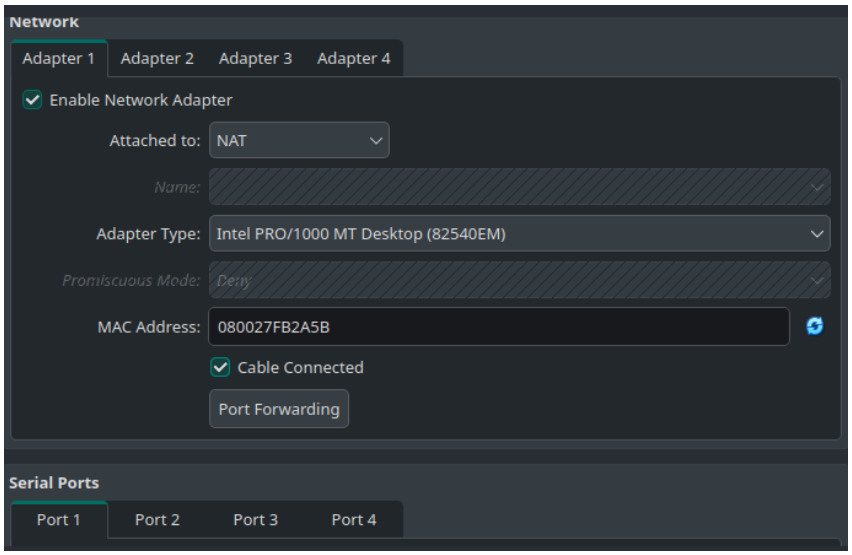

Figure 7: Network configuration

After completely setting up the VM, I was ready to boot it and install Manjaro on it.

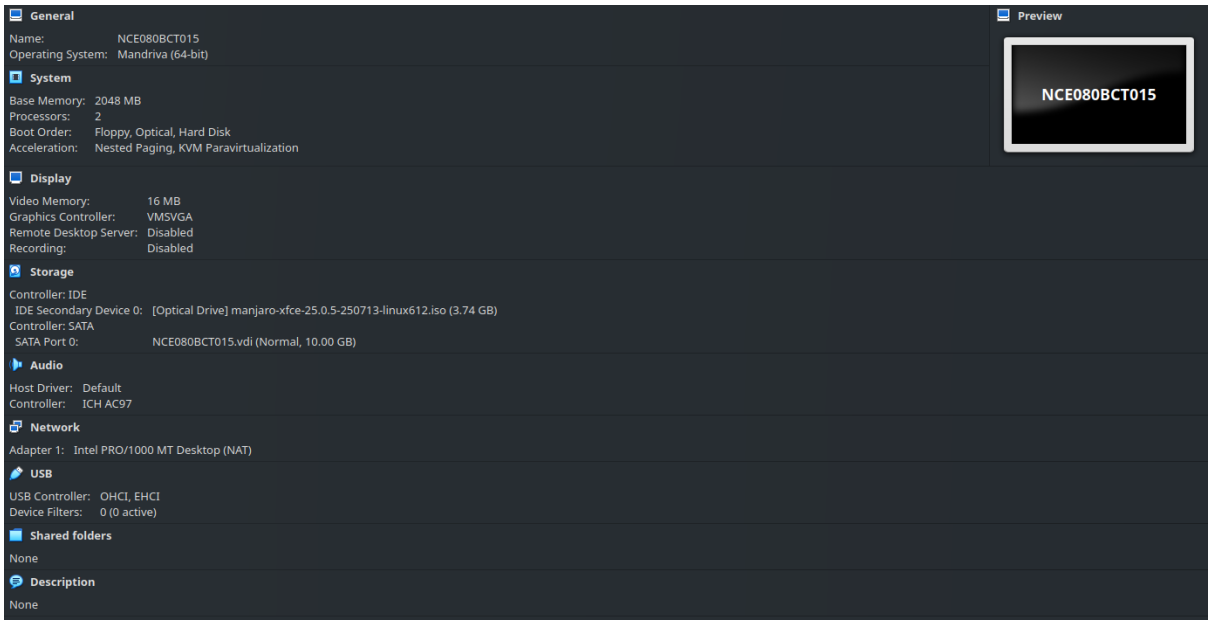

Figure 8: System Summary

4.4 Booting the VM

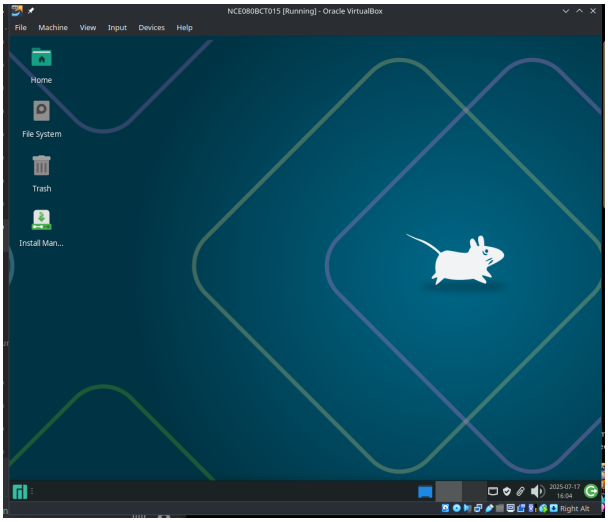

Figure 9: VM Boot

4.5 Installing the OS

4.6 Post-Installation Configuration

4.7 Pausing the VM

4.8 Taking Snapshots

4.9 Exporting the VM

4.10 Importing a VM

4.11 Cloning the VM

5 Observation

6 Results

7 Advantages

8 Conclusion