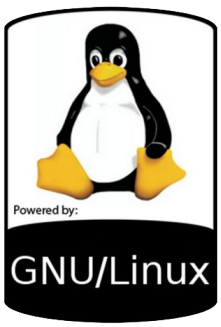
Data Modelling Tools

AUTM08016

Topic 1
Build an Ubuntu Linux Platform on VirtualBox



Dr Diarmuid Ó Briain
Version 1.0 [22 January 2024]

TUS
Ollscoil Teicneolaíochta na Sionainne:
Lár Tíre, An tIarthar Láir
Technological University of the Shannon:
Midlands Midwest

Dr Diarmuid Ó Briain

Linux Version

```
~$ lsb_release -a
No LSB modules are available.
Distributor ID:    Ubuntu
Description:       Ubuntu 22.04.3 LTS
Release:    22.04
Codename:   jammy
```

VirtualBox version

```
~$ virtualboxvm --help | head -1
Oracle VM VirtualBox VM Runner v7.0.12
```

Table of Contents

Table of Figures

1. Introduction

In today's data-driven world, where organisations collect, store, and analyse vast amounts of information, databases and data modelling tools have emerged as essential tools for managing and extracting value from this data. Databases provide a structured repository for storing and organising data, while data modelling tools assist in designing and creating these databases efficiently and effectively. Studying databases and data modelling tools equips learners with the knowledge and skills necessary to design, implement, and maintain efficient and scalable data storage solutions that can power business operations, scientific research, and decision-making across various industries.

1.1 Objectives

At the end of this topic the learner will be able to install Ubuntu Linux in a Virtual Machine (VM) on an Oracle VirtualBox hypervisor.

2. Virtual Machines and Containers

A VM is an operating system that is contained within a software program, called a Hypervisor, on a computer. Like a physical computer, a VM has a Central Processing Unit (CPU), memory, storage disks and network interfaces; however, these are all contained within a program on a physical computer. In cloud computing terms a physical computer is termed bare-metal.

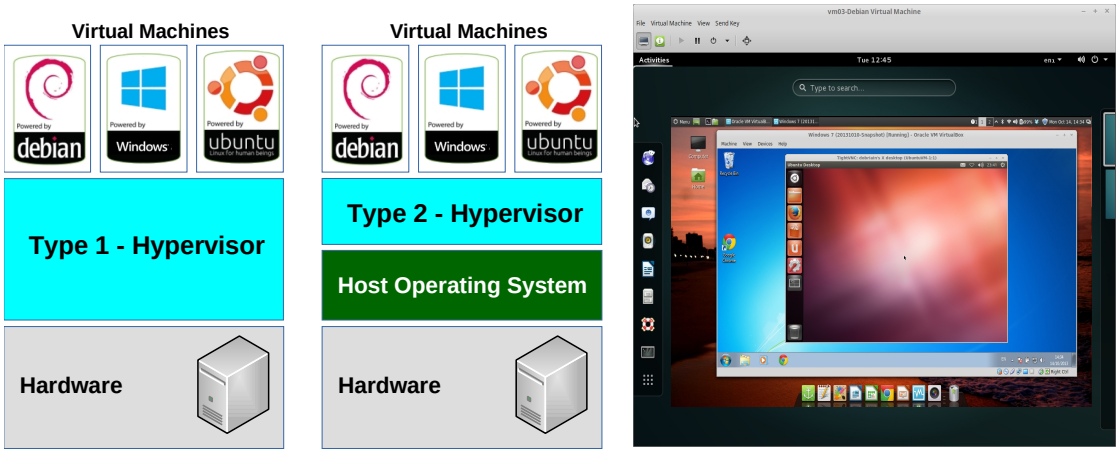


Figure 1: Hypervisors and Virtual Machines

The Hypervisor program manages VMs on the host computer in such a way as to give them access to the hardware by providing virtual hardware to the VM. It is common for modern processors to support the hypervisor by providing isolation from the real hardware. There are two types of Hypervisor, type 1 hypervisors run on bare metal, while Type 2s have an underlying operating system. The example in Figure 1 illustrates a Debian GNU/Linux VM hosting a Linux Mint VM which in turn is hosting a Microsoft Windows VM which in turn is hosting a Ubuntu VM.

3. Containers

An alternative to VMs are containers which provide isolation for applications. Containers have a Container Manager which is often also termed a Hypervisor; however, in this case isolation between the containers is provided by the host operating system through the use of virtual memory. While each VM has its own operating system, containers share many aspects of the host operating system and each container holds the differences from its host. As containers share aspects of the host operating system they enjoy lower overheads than VMs and hardware can therefore support many more containers that would be possible with a hypervisor and VMs.

Consider the diagram to the left side of Figure 2. This mirrors what was illustrated in Figure 1, except that it shows the runtime system of the operating system of each VM and an application or applications running on that VM. The runtime system is the operating system engine that translates application code into machine code for a collection of resources it manages. For the diagram on the right side of Figure 2 each container has its own instance of the runtime system though each runtime is

interfacing with the host operating system and not an operating system running in a VM. In this way the container is in fact an abstraction of the host operating system and applications in each container share the same underlying host operating system.

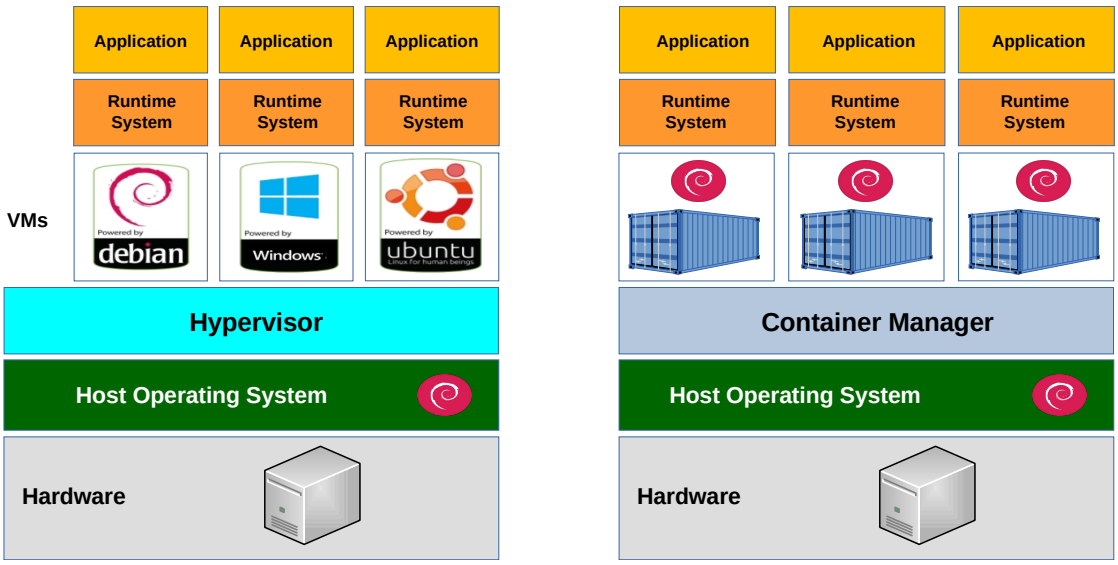


Figure 2: VMs vs Containers

A hybrid approach also exists called para-virtualisation. In this case virtual memory is used to support isolation; however, the VM must have specific device drivers installed that link it to the underlying hardware, via the hypervisor, to access device services.

4. Installation platform

Ubuntu Linux comes in a number of forms for different applications.

4.1 Ubuntu Desktop

Ubuntu desktop is an operating system suitable for installation on a Personal Computer. It can be installed on its own or alongside another operating system such as Microsoft Windows or Apple Mac OS or as a replacement as the sole operating system on the computer. Another option is to install Ubuntu Linux in a VM and that is the option this document will explore further.

4.2 Ubuntu Server

Alongside its parent, Debian GNU/Linux, Ubuntu Server is ideal for deployment in the cloud and data centre. Ubuntu Linux has five years of guaranteed free upgrades. Additionally there is a light build of Ubuntu Server, called Ubuntu Cloud, designed and optimised for rapid deployment on most major clouds such as Amazon AWS, Google Cloud Platform and Microsoft Azure.

4.3 Ubuntu for IoT

This version is a lightweight version ideal for IoT boards such as Raspberry Pi, Intel IoT platforms, Intel NUC, Kernel Virtual Machine (KVM), Qualcomm Dragonboard 410c, Intel IEI TANK 870, Xilinx Evaluation kits & System-on-Modules (SOM).

5. Download Ubuntu Desktop Linux

Download the current stable Ubuntu Desktop ISO file from the URL:

`https://ubuntu.com/download/desktop`

The current stable version (May 2022) is Ubuntu Desktop 22.04 Long Term Support (LTS).

6. Installation Medium

6.1 Installation on a computer

Create Bootable Installation Media by burning the downloaded ISO onto a USB Stick and make it bootable. There are many guides available for all operating systems to explain how to do that.

Ubuntu Linux has a built in Startup Disk Creator. As illustrated in Figure 3, simply search for it in the application search, select the ISO file and the USB Stick and select **Make Startup Disk**.

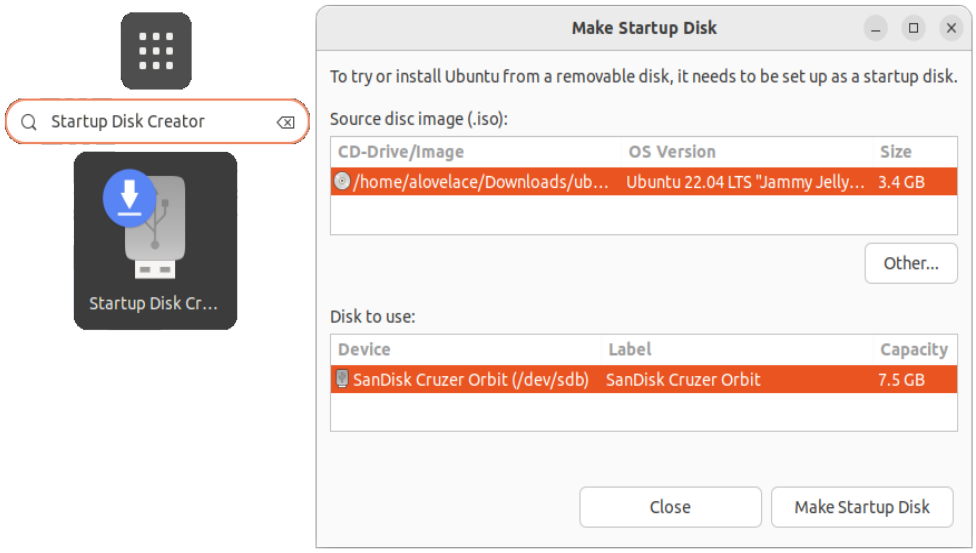


Figure 3: Startup Disk Creator

If an Ubuntu Linux computer is not available, download and install Rufus (**`https://rufus.ie/en`**) . The procedure is similar to that just described.

On the computer that Ubuntu Desktop will be installed on, insert the new bootable USB Stick in one of its USB ports. Boot the computer and Ubuntu Linux will prompt for installation. It may be necessary to enter the computer's Basic Input/Output System (BIOS) settings and change the boot medium from the hard disk to the USB drive.

6.2 Installation on a VirtualBox VM

Oracle VirtualBox is a type-2 hypervisor for x86 virtualisation developed by Oracle Corporation. VirtualBox can be downloaded and installed easily on most operating systems (**`https://www.virtualbox.org/wiki/Downloads`**). The computer where the VirtualBox hypervisor is installed is known as the host and on this, one or more guest operating systems are installed. The hypervisor presents virtual hardware upon which the guest, Virtual Machines (VM) is installed.

Figure 4: VirtualBox Hypervisor with no guest VMs

Select **`New`** to define a new VM for installation.

6.2.1 Add a new VM to host Ubuntu Desktop

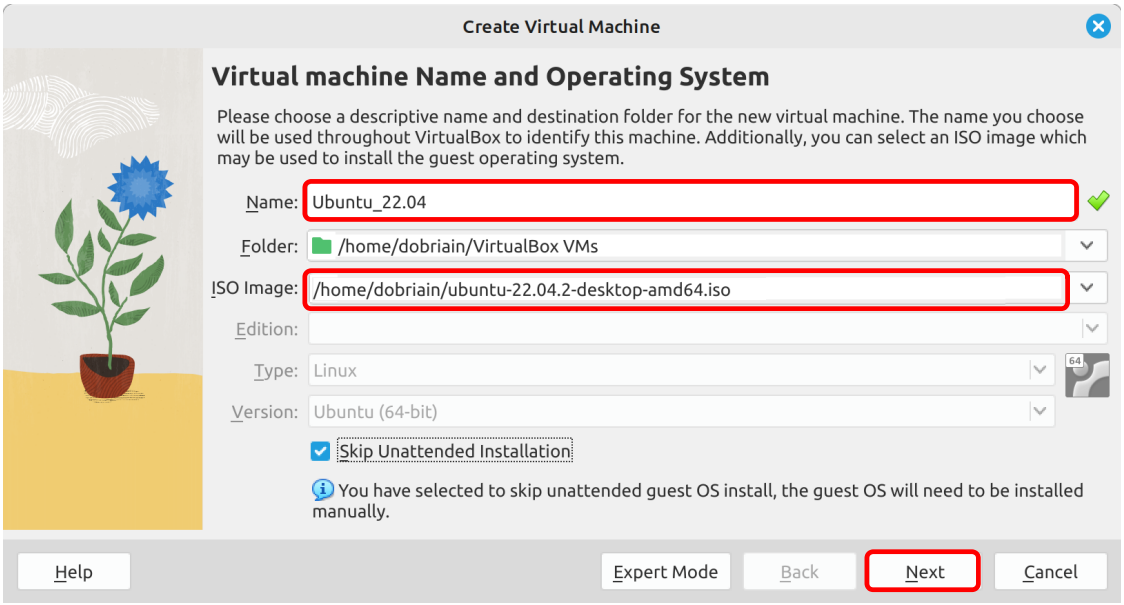


Figure 5: Install a new VM for Ubuntu Desktop

- Specify the **Name:** of the VM, and link to the **.iso** image.
- Select the option **Skip Unattended Installation**.
- Click **Next**.

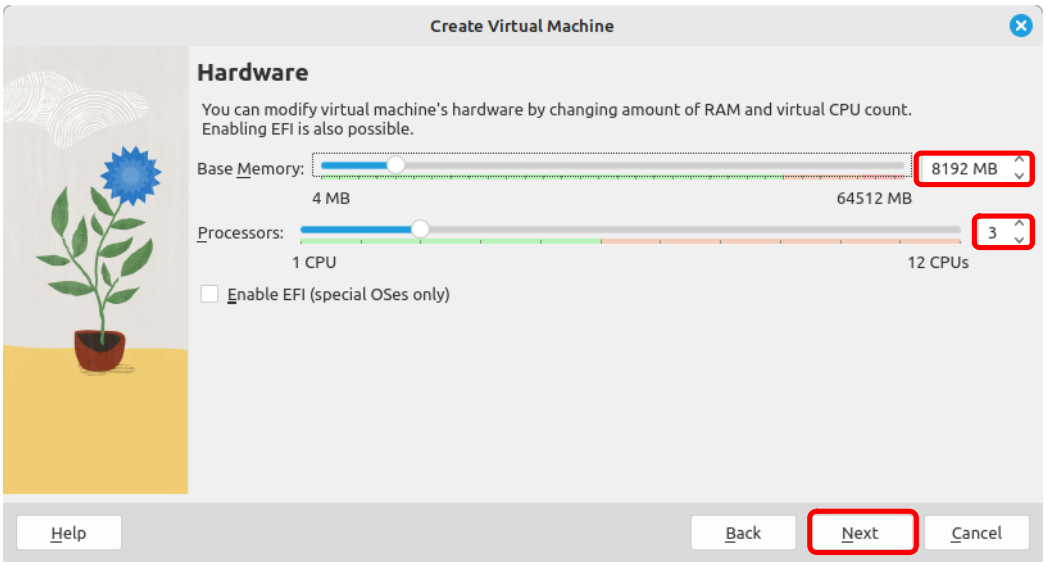


Figure 6: Memory and CPUs

- As illustrated in Figure 6, specify a suitable base memory, for example 8192 MB and 2 or 3 CPUs depending on system availability.
- Click **Next**.

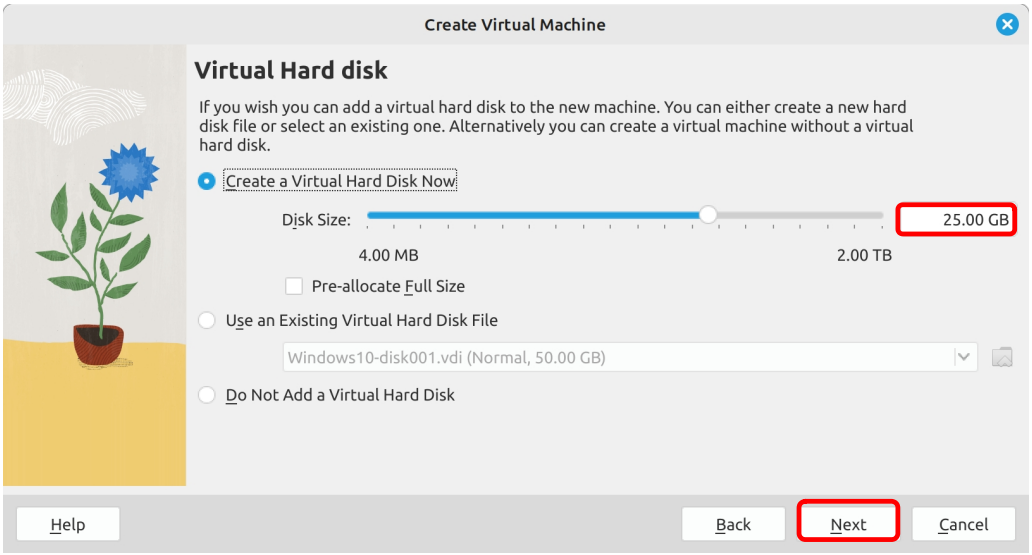


Figure 7: Virtual Harddrive

- Create a Virtual harddrive of 25 GB or more if the hardware has capacity.
- Click **Next**.

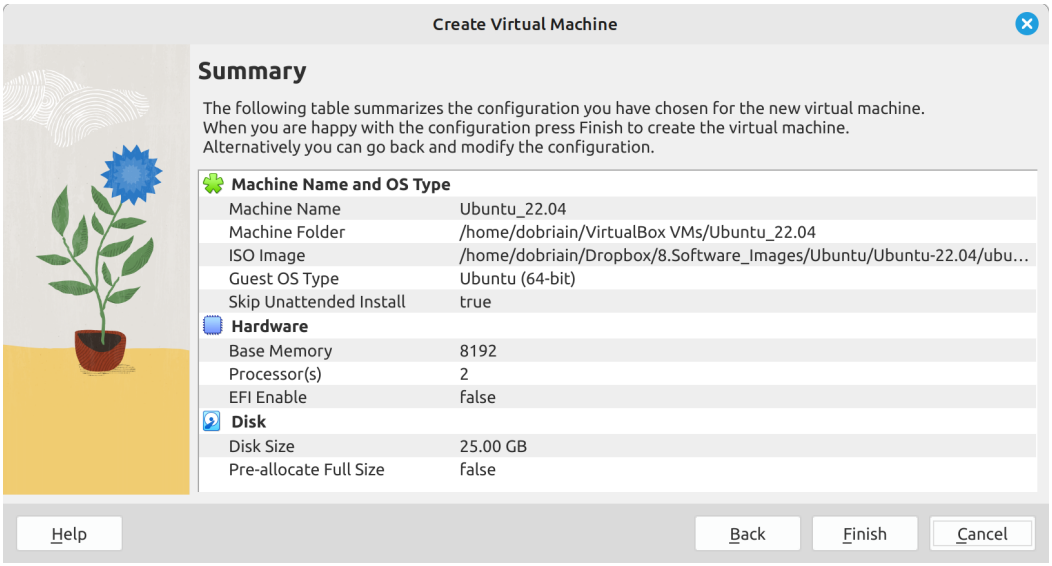


Figure 8: A summary of the system

- A summary of the system defined is presented.
- Click **Finish**.

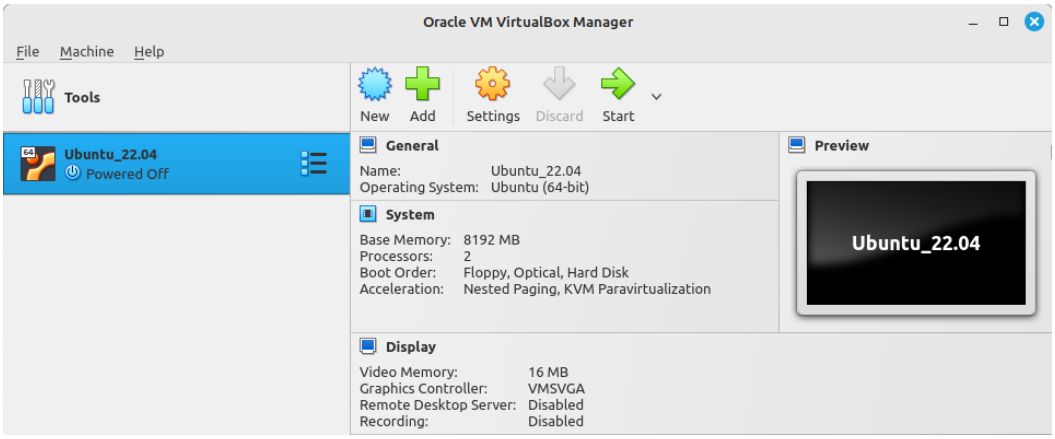


Figure 9: VM created and read for software installation

- Highlight the newly installed VM.
- Click the **Start** icon.

Figure 10: Boot options

- Select **Try or Install Ubuntu**.

7. Install Ubuntu Desktop

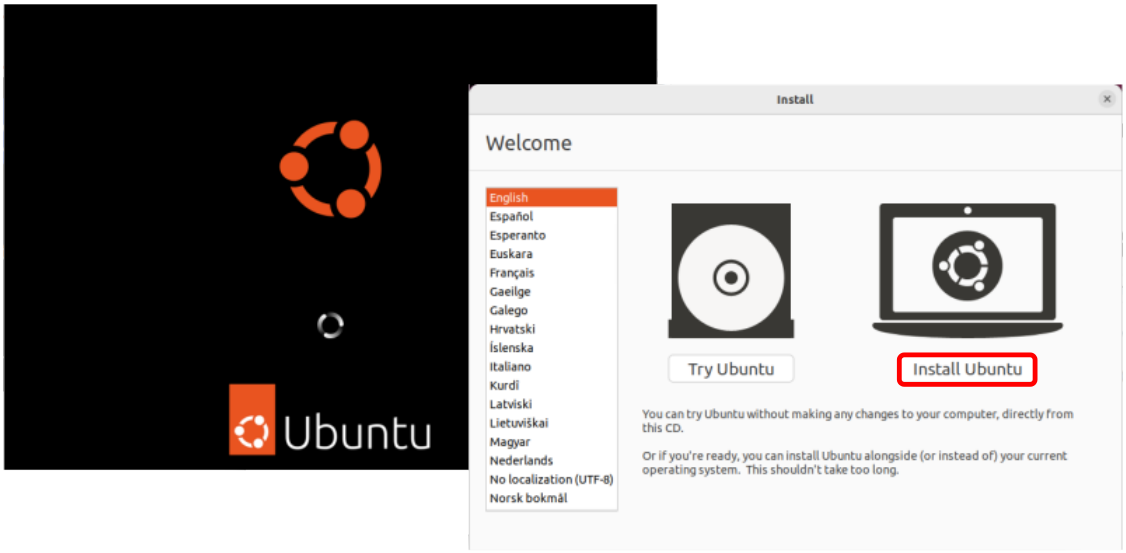


Figure 11: Ubuntu Desktop installation

No matter whether the installation is on a computer or a VM on a hypervisor, the installation will arrive to a welcome screen.

- Select **Install Ubuntu**.
- **Keyboard layout**:
  - Select preferred keyboard, I.e. **English UK (Intl, with dead keys)**.
  - Click **Continue**.
- **Updates and other software**:
  - There two options, **Normal** and **Minimal** Installation. In Normal Installation, all Graphical User Interface (GUI) related applications will be installed whereas with minimal installation only basic applications are installed.
  - Select **Normal Installation** and **Install 3rd Party Software**.
  - Click **Continue**.
- **Installation Type**:
  - Select **Erase Disk and install Ubuntu**.
  - **Note**: If this installation is on a computer, shared with another operating system then it will be necessary to create partitions to accommodate both operating systems. This is outside the scope of this introductory module.
  - Select **Install Now**.
- **Write the changes to disks:**
  - Click **Continue**.

The base operating system will take some time to install at this point.

7.1 Configure the Operating System

A Map of the world will be presented.

- **Where are you?**:
  - Select location, i.e. **Dublin**.
  - Click **Continue**.
- **Who are you?**:
  - Fill-out the details of the primary user.
    - Your name: **Ada Lovelace**
    - Your computer's name: **ada-VirtualBox**
    - Pick a username: **ada**
    - Choose a password: **ada_pass**
    - Confirm your password: **ada_pass**
    - Select **Require my password to login**.
    - Click **Continue**.

The software will take some time to install at this point.

Ubuntu Desktop will now install on the VM.

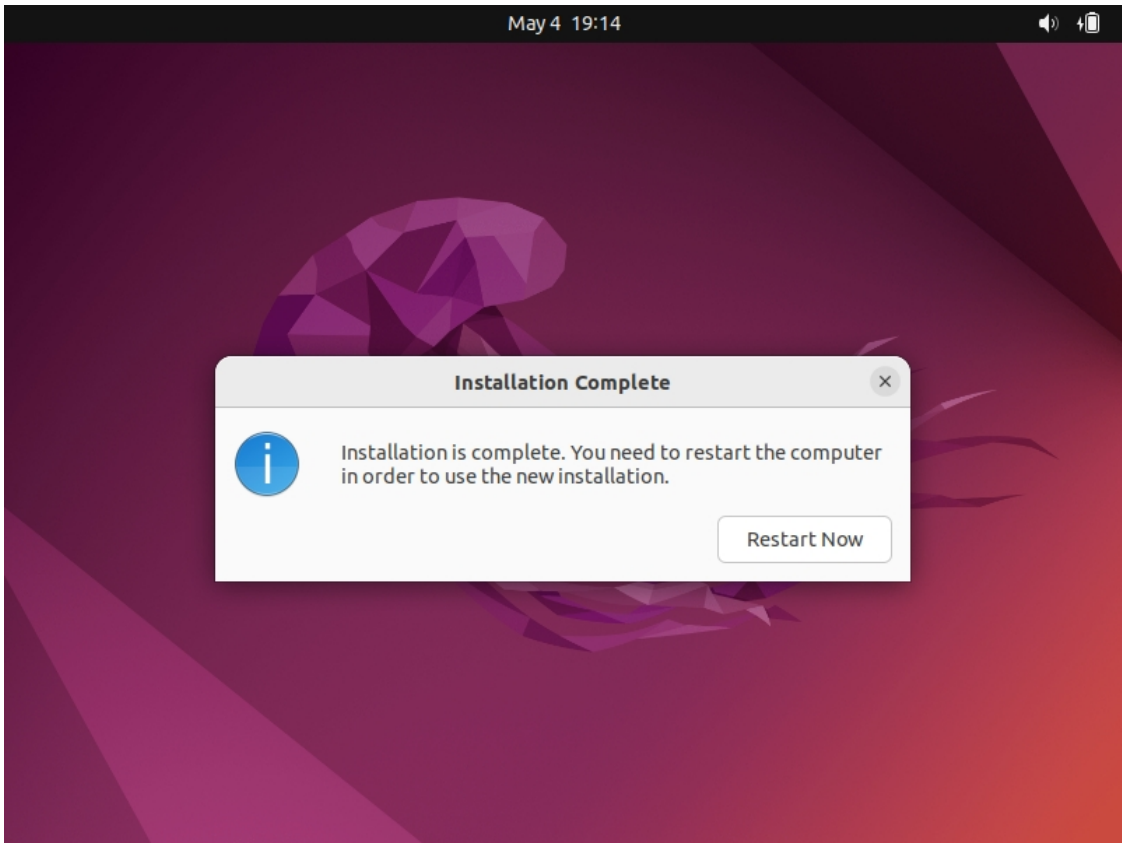


Figure 12: Ubuntu Desktop installation complete

When Ubuntu Linux has installed, restart the VM and login once is has rebooted. If the installation was made on computer hardware then do not forget to remove the bootable USB stick and change the boot medium from USB back to hard disk from BIOS settings if you found that you needed to change them.

Note that the created user will have administrative rights. On GNU/Linux such rights are called **`root`** rights and they are achieved through the Super User do (**`sudo`**) command.

8. Post installation

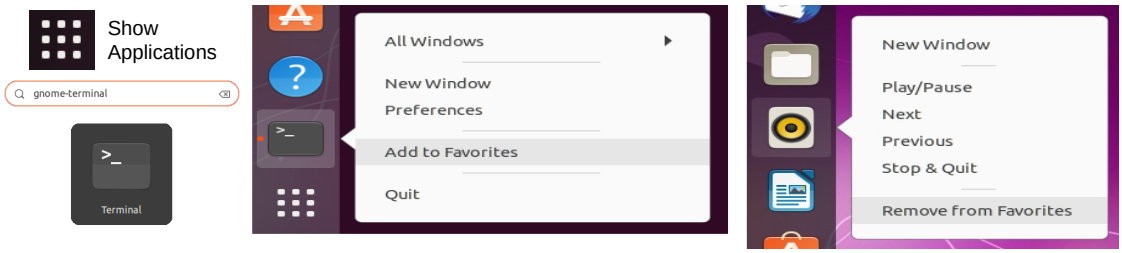


Figure 13: Add/Remove from Favourites

On the left of the desktop is the dock. It gives access to the applications installed on the operating system. The icon on the bottom left of the dock shows the applications.

8.1 Add a terminal to the dock

As illustrated in Figure 13 search for **`gnome-terminal`**, run it and while it is running right click on the temporary icon that appears in the **dock** and select **Add to Favorites**.

8.2 Removing unnecessary items from the dock

Also illustrated in Figure 13 is the removal of **Rhythmbox** from the **dock**. Right click on the icon and select **Remove to Favorites**.

8.3 Update the Operating System

In common with most GNU/Linux distributions, Ubuntu hosts a repository of applications which can be accessed and updated at any time. In fact Ubuntu will offer you reminders from time to time. These updates can be carried out through the Ubuntu Software Centre; however, it is much faster to complete this task through the terminal.

Run the terminal by selecting the icon you recently added to the **dock** and update the package information from all configured sources. When this is complete install available upgrades of all packages currently installed on the system from those sources.

```
ada:~$ sudo apt update
Fetched 369 kB in 1s (382 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.

ada:~$ sudo apt -y upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

8.4 Add some essential packages

gcc – GNU project C and C++ compiler. This is essential for compiling source code.

make – Utility will determine automatically which pieces of a large program need to be recompiled, and issue the commands to recompile them.

openssh-server – OpenSSH is a suite of secure networking utilities based on the Secure SHell (SSH) protocol, which provides a secure channel over an unsecured network in a client–server architecture.

```
ada:~$ sudo apt -y install gcc make openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  gcc-multilib autoconf automake libtool flex bison gcc-doc make-doc
  molly-guard monkeysphere ssh-askpass
The following NEW packages will be installed:
  gcc make openssh-server
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 550 kB of archives.
After this operation, 1,991 kB of additional disk space will be used.
Preconfiguring packages …
Setting up openssh-server (1:8.3p1-1) ...
Setting up gcc (4:10.2.0-1ubuntu1) ...
Setting up make (4.3-4ubuntu1) ...
Processing triggers for systemd (246.6-1ubuntu1) ...
Processing triggers for man-db (2.9.3-2) ...
Processing triggers for ufw (0.36-7) ...
```

8.5 Add Secure Shell keys

ssh-keygen generates, manages and converts authentication keys for Secure Shell (SSH), an essential tool for connectivity to other devices or to secure connections from other devices.

- **-t** Specifies the type of key to create, in this case a Rivest–Shamir–Adleman (RSA) key pair includes a public and a private key. The RSA private key is used to generate digital signatures, and the RSA public key is used to verify digital signatures.
- **-b** Specifies the number of bits in the key to create. For RSA keys, the minimum size is 1024 bits and the default is 3072 bits.

```
ada:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ada/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ada/.ssh/id_rsa
Your public key has been saved in /home/ada/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:Pm4KoPJHgvlnVsJ9AuBpjVNDLMoA5o7HBHkQBkiH6gI ub-g3-20-04@ubuntu
The key's randomart image is:
+---[RSA 4096]----+
|@B.+.            |
|Bo= +            |
|+=.B .           |
|E+* o            |
|+=+o o  S        |
|++..+ +..        |
|+. o.o oo        |
|... =. ...       |
|  .=  .o.        |
+----[SHA256]-----+
```

8.6 VirtualBox Guest Additions

This only applies to Ubuntu Linux installed in a VirtualBox VM. Guest Additions are designed to be installed ***inside*** a VM after the guest operating system has been installed. They consist of device drivers and system applications that optimise the guest operating system for better performance and usability.

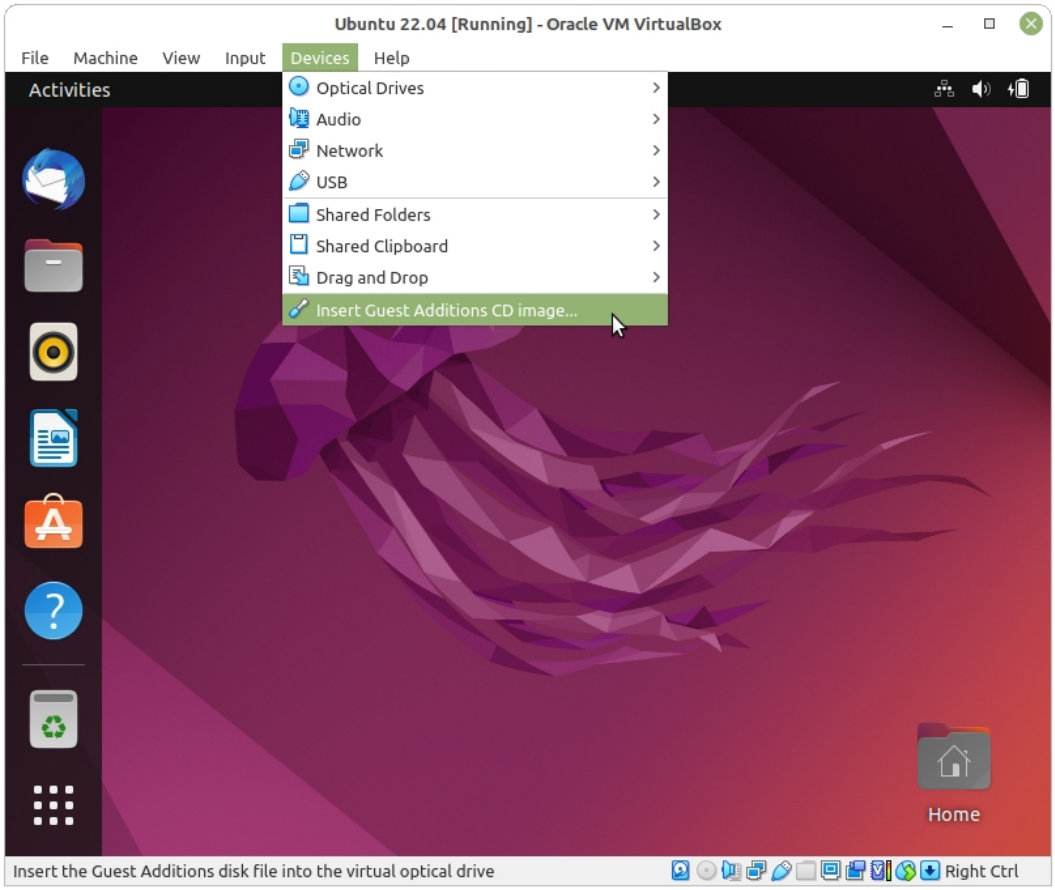


Figure 14: VirtualBox Guest Additions

VirtualBox provides a virtual disk with the Guest Additions software on it. From the guest VMs perspective it is as if a CD was installed in its CD slot.

Insert this **Guest Additions CD** image by selecting the option under **Devices** in the VM menu. That will then mount the virtual CD at **`/media/ada`** on the Ubuntu Linux filesystem.

Change to this directory and run the **`autorun.sh`** program as shown below. After they have installed, rerun the VM to enable the additions.

```
ada:~$ cd /media/ada/Vbox_Gas_6.1.10/

ada:/media/ada/Vbox_Gas_6.1.10~$ sudo ./autorun.sh
Verifying archive integrity... All good.
Uncompressing VirtualBox 6.1.32 Guest Additions for Linux........
VirtualBox Guest Additions installer
Removing installed version 6.1.32 of VirtualBox Guest Additions...
Copying additional installer modules ...
Installing additional modules ...
VirtualBox Guest Additions: Starting.
VirtualBox Guest Additions: Building the VirtualBox Guest Additions
kernel
modules.  This may take a while.
VirtualBox Guest Additions: To build modules for other installed
kernels, run
VirtualBox Guest Additions:   /sbin/rcvboxadd quicksetup <version>
VirtualBox Guest Additions: or
VirtualBox Guest Additions:   /sbin/rcvboxadd quicksetup all
VirtualBox Guest Additions: Building the modules for kernel 5.8.0-36-
generic.
VirtualBox Guest Additions: Running kernel modules will not be replaced
until
the system is restarted
Press Return to close this window...

ada:/media/ada/Vbox_Gas_6.1.10~$ sudo reboot now
```

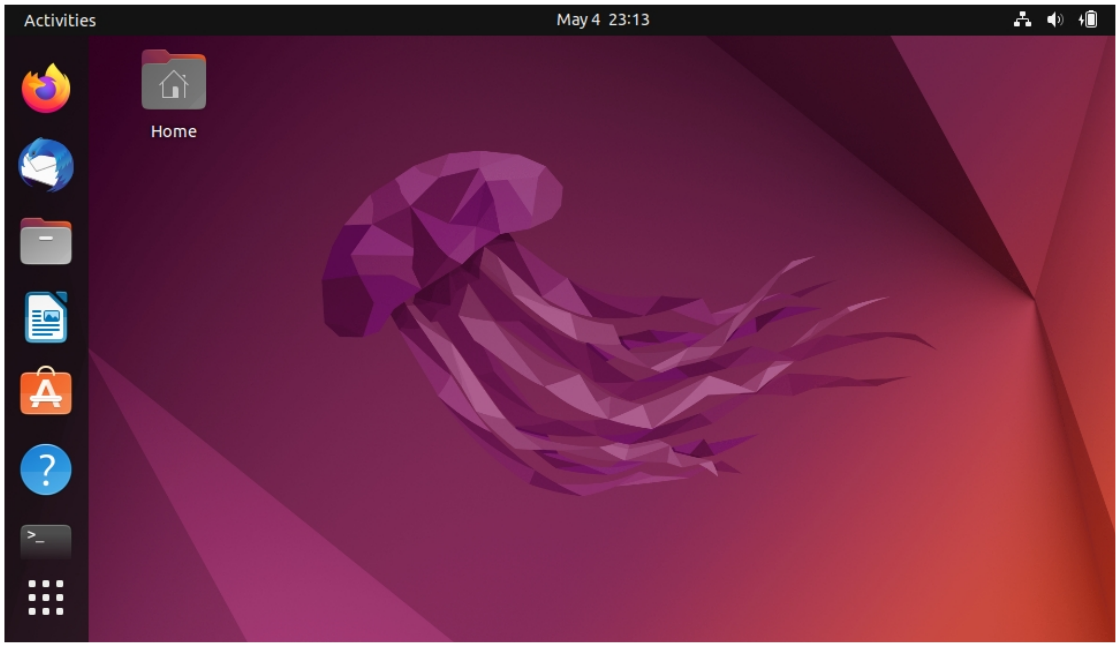


Figure 15: Completed Ubuntu Linux installation

9. Exercise #1

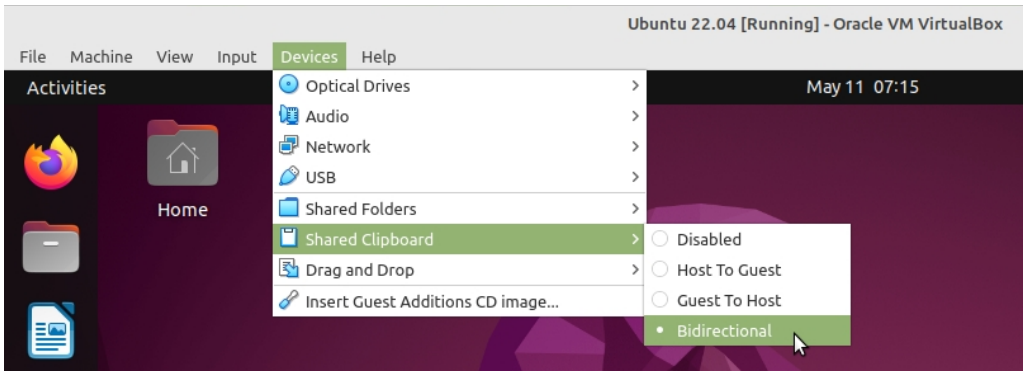


Figure 16: Shared Clipboard/Drag and Drop

- Install Ubuntu Desktop version 22.04, Jammy Jellyfish, in a VirtualBox Virtual Machine as described.
  - Add the Gnome **`text-editor`** and **`calculator`** to the Dock
  - Set the VirtualBox **Shared Clipboard** to **Bidirectional**
  - Set the VirtualBox **Drag and Drop** to **Bidirectional**
  - What do these additions do?

Notes: