

Ubuntu 20.04 Essentials

Ubuntu 20.04 Essentials

ISBN-13: 978-1-951442-05-7

© 2020 Neil Smyth / Payload Media, Inc. All Rights Reserved.

This book is provided for personal use only. Unauthorized use, reproduction and/or distribution strictly prohibited. All rights reserved.

The content of this book is provided for informational purposes only. Neither the publisher nor the author offers any warranties or representation, express or implied, with regard to the accuracy of information contained in this book, nor do they accept any liability for any loss or damage arising from any errors or omissions.

This book contains trademarked terms that are used solely for editorial purposes and to the benefit of the respective trademark owner. The terms used within this book are not intended as infringement of any trademarks.

Rev: 1.0

Table of Contents

1. Introduction	1
1.1 Superuser Conventions.....	1
1.2 Opening a Terminal Window	2
1.3 Editing Files	3
1.4 Feedback.....	4
1.5 Errata.....	5
2. A Brief History of Linux.....	7
2.1 What exactly is Linux?.....	7
2.2 UNIX Origins	7
2.3 Who Created Linux?	7
2.4 The History of Ubuntu	8
2.5 What does the word “Ubuntu” Mean?	8
2.6 Summary	8
3. Installing Ubuntu on a Clean Disk Drive	9
3.1 Ubuntu Installation Options.....	9
3.2 Server vs. Desktop Editions	10
3.3 Obtaining the Ubuntu Installation Media	11
3.4 Writing the ISO Installation Image to a USB Drive.....	11
3.4.1 Linux.....	11
3.4.2 macOS	12
3.4.3 Windows	13
3.5 Booting from the Ubuntu USB Image.....	14
3.6 Installing Ubuntu	15
3.7 Accessing the Ubuntu Desktop	20
3.8 Installing Updates.....	21
3.9 Displaying Boot Messages.....	22
3.10 Summary	23
4. Installing Ubuntu with the Network Installer	25
4.1 Network Installer Advantages.....	25
4.2 Obtaining the Network Installer Image.....	25
4.3 Booting from the Installer Image	25
4.4 Performing the Installation	26
4.5 Disk Partitioning	28
4.6 Software Collection Selection	30
4.7 Installing Software Collections After System Setup.....	31
4.8 Summary	31

Table of Contents

5. Dual Booting Ubuntu with Windows	33
5.1 Beginning the Ubuntu Installation	33
5.2 Booting Ubuntu for the First Time.....	38
5.3 Changing the Default Boot Option.....	39
5.4 Accessing the Windows Partition from the Command-line.....	40
5.5 Accessing the Windows Partition from the Desktop.....	41
5.6 Summary	44
6. Allocating Windows Disk Partitions to Ubuntu	45
6.1 Unmounting the Windows Partition.....	45
6.2 Deleting the Windows Partitions from the Disk	45
6.3 Formatting the Unallocated Disk Partition	48
6.4 Mounting the New Partition	48
6.5 Editing the Boot Menu	49
6.6 Using the GNOME Disks Utility.....	49
6.7 Summary	54
7. A Guided Tour of the GNOME 3 Desktop	55
7.1 Installing the GNOME Desktop.....	55
7.2 An Overview of the GNOME 3 Desktop	55
7.3 Launching Activities	57
7.4 Managing Windows	59
7.5 Using Workspaces	59
7.6 Calendar and Notifications	60
7.7 Desktop Settings	61
7.8 Customizing the Dash	62
7.9 Switching to Dark Mode.....	62
7.10 Installing Ubuntu Software	63
7.11 Beyond Basic Customization	64
7.12 Summary	65
8. An Overview of the Ubuntu Cockpit Web Interface	67
8.1 An Overview of Cockpit.....	67
8.2 Installing and Enabling Cockpit.....	68
8.3 Accessing Cockpit	68
8.4 Overview	69
8.5 Logs	70
8.6 Storage.....	70
8.7 Networking.....	71
8.8 Accounts	71
8.9 Services	71
8.10 Applications	72
8.11 Virtual Machines.....	72

8.12 Software Updates.....	73
8.13 Terminal	73
8.14 Connecting to Multiple Servers	74
8.15 Summary	76
9. Using the Bash Shell on Ubuntu.....	77
9.1 What is a Shell?	77
9.2 Gaining Access to the Shell	77
9.3 Entering Commands at the Prompt.....	78
9.4 Getting Information about a Command	78
9.5 Bash Command-line Editing.....	78
9.6 Working with the Shell History.....	79
9.7 Filename Shorthand.....	80
9.8 Filename and Path Completion.....	80
9.9 Input and Output Redirection	80
9.10 Working with Pipes in the Bash Shell.....	81
9.11 Configuring Aliases	81
9.12 Environment Variables	82
9.13 Writing Shell Scripts	83
9.14 Summary	84
10. Managing Ubuntu Users and Groups	85
10.1 User Management from the Command-line	85
10.2 User Management with Cockpit.....	87
10.3 User Management using the Settings App.....	88
10.4 Summary	90
11. Managing Ubuntu systemd Units.....	91
11.1 Understanding Ubuntu systemd Targets	91
11.2 Understanding Ubuntu systemd Services.....	91
11.3 Ubuntu systemd Target Descriptions	91
11.4 Identifying and Configuring the Default Target	93
11.5 Understanding systemd Units and Unit Types.....	94
11.6 Dynamically Changing the Current Target	94
11.7 Enabling, Disabling and Masking systemd Units	95
11.8 Working with systemd Units in Cockpit.....	96
11.9 Summary	97
12. Ubuntu Software Package Management and Updates.....	99
12.1 Repositories.....	99
12.2 Managing Repositories with Software & Updates	100
12.3 Managing Packages with APT	102
12.4 Performing Updates.....	103
12.5 Enabling Automatic Updates.....	104

Table of Contents

12.6 Enabling Livepatch.....	105
12.7 Summary	107
13. Ubuntu Snap Package Management	109
13.1 Managing Software with Snap	109
13.2 Basic Snap Commands	110
13.3 Working with Snap Channels	112
13.4 Snap Refresh Schedule.....	113
13.5 Snap Services	114
13.6 Summary	115
14. Ubuntu Network Management	117
14.1 An Introduction to NetworkManager	117
14.2 Installing and Enabling NetworkManager.....	118
14.3 Basic nmcli Commands.....	118
14.4 Working with Connection Profiles	122
14.5 Interactive Editing.....	125
14.6 Configuring NetworkManager Permissions.....	126
14.7 Summary	127
15. Ubuntu Firewall Basics.....	129
15.1 Understanding Ports and Services	129
15.2 Securing Ports and Services.....	129
15.3 Ubuntu Services and iptables Rules.....	130
15.4 Well Known Ports and Services.....	131
15.5 Summary	136
16. Using gufw and ufw to Configure an Ubuntu Firewall	137
16.1 An Overview of gufw and ufw	137
16.2 Installing gufw on Ubuntu	137
16.3 Running and Enabling gufw.....	137
16.4 Creating a New Profile.....	138
16.5 Adding Preconfigured Firewall Rules.....	140
16.6 Adding Simple Firewall Rules.....	141
16.7 Adding Advanced Rules	142
16.8 Configuring the Firewall from the Command Line using ufw	143
16.9 Summary	145
17. Basic Ubuntu Firewall Configuration with firewalld	147
17.1 An Introduction to firewalld.....	147
17.1.1 Zones	147
17.1.2 Interfaces.....	149
17.1.3 Services.....	149
17.1.4 Ports.....	149

17.2	Checking firewalld Status	149
17.3	Configuring Firewall Rules with firewall-cmd.....	150
17.3.1	Identifying and Changing the Default Zone.....	150
17.3.2	Displaying Zone Information	150
17.3.3	Adding and Removing Zone Services.....	151
17.3.4	Working with Port-based Rules.....	152
17.3.5	Creating a New Zone.....	152
17.3.6	Changing Zone/Interface Assignments.....	152
17.3.7	Masquerading	152
17.3.8	Adding ICMP Rules	153
17.3.9	Implementing Port Forwarding.....	153
17.4	Managing firewalld using firewall-config	154
17.5	Summary	155
18.	Configuring SSH Key-based Authentication on Ubuntu	157
18.1	An Overview of Secure Shell (SSH).....	157
18.2	SSH Key-based Authentication	157
18.3	Setting Up Key-based Authentication	158
18.4	Installing and Starting the SSH Service.....	158
18.5	SSH Key-based Authentication from Linux and macOS Clients.....	158
18.6	Managing Multiple Keys	160
18.7	SSH Key-based Authentication from Windows 10 Clients	161
18.8	SSH Key-based Authentication using PuTTY	163
18.9	Generating a Private Key with PuTTYgen	164
18.10	Installing the Public Key for a Google Cloud Instance.....	165
18.11	Summary	166
19.	Ubuntu Remote Desktop Access with Vino	169
19.1	Remote Desktop Access Types	169
19.2	Secure and Insecure Remote Desktop Access	169
19.3	Enabling Remote Desktop Access on Ubuntu.....	170
19.4	Connecting to the Shared Desktop.....	171
19.5	Connecting from Non-Linux Clients	173
19.6	Establishing a Secure Remote Desktop Session	175
19.7	Establishing a Secure Tunnel on Windows using PuTTY	176
19.8	Summary	177
20.	Ubuntu Remote Desktop Access with VNC.....	179
20.1	Installing the GNOME Desktop Environment	179
20.2	Installing VNC on Ubuntu	179
20.3	Configuring the VNC Server.....	180
20.4	Starting the VNC Server	180
20.5	Connecting to a VNC Server.....	181

Table of Contents

20.6 Summary	181
21. Displaying Ubuntu Applications Remotely (X11 Forwarding)	183
21.1 Requirements for Remotely Displaying Ubuntu Applications	183
21.2 Remotely Displaying an Ubuntu Application	184
21.3 Trusted X11 Forwarding	184
21.4 Compressed X11 Forwarding	184
21.5 Displaying Remote Ubuntu Apps on Windows	185
21.6 Summary	188
22. Using NFS to Share Ubuntu Files with Remote Systems	189
22.1 Ensuring NFS Services are running on Ubuntu	189
22.2 Configuring the Ubuntu Firewall to Allow NFS Traffic	189
22.3 Specifying the Folders to be Shared	190
22.4 Accessing Shared Ubuntu Folders	191
22.5 Mounting an NFS Filesystem on System Startup	191
22.6 Unmounting an NFS Mount Point	191
22.7 Accessing NFS Filesystems in Cockpit	192
22.8 Summary	193
23. Sharing Files between Ubuntu and Windows Systems with Samba	195
23.1 Accessing Windows Resources from the GNOME Desktop	195
23.2 Samba and Samba Client	196
23.3 Installing Samba on an Ubuntu System	196
23.4 Configuring the Ubuntu Firewall to Enable Samba	196
23.5 Configuring the <i>smb.conf</i> File	197
23.5.1 Configuring the [global] Section	197
23.5.2 Configuring a Shared Resource	198
23.5.3 Removing Unnecessary Shares	198
23.6 Creating a Samba User	198
23.7 Testing the <i>smb.conf</i> File	199
23.8 Starting the Samba and NetBIOS Name Services	200
23.9 Accessing Samba Shares	201
23.10 Accessing Windows Shares from Ubuntu	202
23.11 Summary	204
24. An Overview of Virtualization Techniques	205
24.1 Guest Operating System Virtualization	205
24.2 Hypervisor Virtualization	206
24.2.1 Paravirtualization	207
24.2.2 Full Virtualization	208
24.2.3 Hardware Virtualization	208
24.3 Virtual Machine Networking	209
24.4 Summary	209

25. Installing KVM Virtualization on Ubuntu	211
25.1 An Overview of KVM	211
25.2 KVM Hardware Requirements.....	211
25.3 Preparing Ubuntu for KVM Virtualization	212
25.4 Verifying the KVM Installation.....	212
25.5 Summary	214
26. Creating KVM Virtual Machines using Cockpit and virt-manager.....	215
26.1 Installing the Cockpit Virtual Machines Module	215
26.2 Creating a Virtual Machine in Cockpit.....	215
26.3 Starting the Installation	217
26.4 Working with Storage Volumes and Storage Pools.....	219
26.5 Creating a Virtual Machine using virt-manager	222
26.6 Starting the Virtual Machine Manager.....	222
26.7 Configuring the KVM Virtual System	223
26.8 Starting the KVM Virtual Machine	226
26.9 Summary	227
27. Creating KVM Virtual Machines with virt-install and virsh	229
27.1 Running virt-install to build a KVM Guest System	229
27.2 An Example Ubuntu virt-install Command	229
27.3 Starting and Stopping a Virtual Machine from the Command-Line	230
27.4 Creating a Virtual Machine from a Configuration File.....	231
27.5 Summary	231
28. Creating an Ubuntu KVM Networked Bridge Interface	233
28.1 Identifying the Network Management System.....	233
28.2 Getting the Netplan Network Settings	234
28.3 Creating a Netplan Network Bridge	235
28.4 Getting the Current Network Manager Settings.....	237
28.5 Creating a Network Manager Bridge from the Command-Line	239
28.6 Declaring the KVM Bridged Network	240
28.7 Using a Bridge Network in a Virtual Machine	241
28.8 Creating a Bridge Network using nm-connection-editor.....	242
28.9 Summary	245
29. Managing KVM using the virsh Command-Line Tool.....	247
29.1 The virsh Shell and Command-Line.....	247
29.2 Listing Guest System Status	248
29.3 Starting a Guest System	249
29.4 Shutting Down a Guest System	249
29.5 Suspending and Resuming a Guest System	249
29.6 Saving and Restoring Guest Systems	249

Table of Contents

29.7 Rebooting a Guest System	250
29.8 Configuring the Memory Assigned to a Guest OS	250
29.9 Summary	250
30. An Introduction to Linux Containers.....	251
30.1 Linux Containers and Kernel Sharing.....	251
30.2 Container Uses and Advantages.....	252
30.3 Ubuntu Container Tools	253
30.4 The Docker Registry	253
30.5 Container Networking.....	254
30.6 Summary	254
31. Working with Containers on Ubuntu	255
31.1 Installing the Container Tools.....	255
31.2 Pulling a Container Image	255
31.3 Running the Image in a Container	257
31.4 Managing a Container	258
31.5 Saving a Container to an Image	259
31.6 Removing an Image from Local Storage.....	259
31.7 Removing Containers.....	259
31.8 Building a Container with Buildah.....	260
31.9 Summary	260
32. Setting Up an Ubuntu Web Server	261
32.1 Requirements for Configuring an Ubuntu Web Server	261
32.2 Installing the Apache Web Server Packages	261
32.3 Configuring the Firewall	262
32.4 Port Forwarding	262
32.5 Starting the Apache Web Server	262
32.6 Testing the Web Server	263
32.7 Configuring the Apache Web Server for Your Domain	263
32.8 The Basics of a Secure Web Site.....	265
32.9 Configuring Apache for HTTPS	266
32.10 Obtaining an SSL Certificate	266
32.11 Summary	269
33. Configuring an Ubuntu Postfix Email Server.....	271
33.1 The structure of the Email System	271
33.1.1 Mail User Agent	271
33.1.2 Mail Transfer Agent	271
33.1.3 Mail Delivery Agent	272
33.1.4 SMTP	272
33.1.5 SMTP Relay	272
33.2 Configuring an Ubuntu Email Server	272

33.3 Postfix Pre-Installation Steps	272
33.4 Firewall/Router Configuration.....	273
33.5 Installing Postfix on Ubuntu.....	273
33.6 Configuring Postfix	274
33.7 Configuring DNS MX Records	276
33.8 Starting Postfix on an Ubuntu System.....	276
33.9 Testing Postfix.....	277
33.10 Sending Mail via an SMTP Relay Server.....	277
33.11 Summary	278
34. Adding a New Disk Drive to an Ubuntu System.....	279
34.1 Mounted File Systems or Logical Volumes	279
34.2 Finding the New Hard Drive	279
34.3 Creating Linux Partitions	280
34.4 Creating a File System on a Disk Partition	281
34.5 An Overview of Journaled File Systems.....	281
34.6 Mounting a File System	282
34.7 Configuring Ubuntu to Automatically Mount a File System	283
34.8 Adding a Disk Using Cockpit.....	283
34.9 Summary	285
35. Adding a New Disk to an Ubuntu Volume Group and Logical Volume	287
35.1 An Overview of Logical Volume Management (LVM)	287
35.1.1 Volume Group (VG).....	287
35.1.2 Physical Volume (PV)	287
35.1.3 Logical Volume (LV)	288
35.1.4 Physical Extent (PE)	288
35.1.5 Logical Extent (LE)	288
35.2 Getting Information about Logical Volumes	288
35.3 Adding Additional Space to a Volume Group from the Command-Line	290
35.4 Summary	292
36. Adding and Managing Ubuntu Swap Space.....	293
36.1 What is Swap Space?.....	293
36.2 Recommended Swap Space for Ubuntu.....	293
36.3 Identifying Current Swap Space Usage	294
36.4 Adding a Swap File to an Ubuntu System.....	294
36.5 Adding Swap as a Partition.....	295
36.6 Adding Space to an Ubuntu LVM Swap Volume.....	295
36.7 Adding Swap Space to the Volume Group.....	297
36.8 Summary	298
37. Ubuntu System and Process Monitoring	299
37.1 Managing Processes.....	299

Table of Contents

37.2 Real-time System Monitoring with htop.....	303
37.3 Command-Line Disk and Swap Space Monitoring.....	304
37.4 Summary	305
Index	307

1. Introduction

Ubuntu is arguably one of the most highly regarded and widely used Linux distributions available today. Praised both for its ease of use and reliability, Ubuntu also has a loyal following of Linux users and an active community of developers.

Ubuntu 20.04 Essentials is designed to provide detailed information on the installation, use and administration of the Ubuntu 20.04 distribution. For beginners, the book covers topics such as operating system installation, the basics of the GNOME desktop environment, configuring email and web servers and installing packages and system updates. Additional installation topics such as dual booting with Microsoft Windows are also covered, together with all important security topics such as configuring a firewall and user and group administration.

For the experienced user, topics such as remote desktop access, the Cockpit web interface, logical volume management (LVM), disk partitioning, swap management, KVM virtualization, Secure Shell (SSH), Linux Containers and file sharing using both Samba and NFS are covered in detail to provide a thorough overview of this enterprise class operating system.

1.1 Superuser Conventions

Ubuntu, in common with Linux in general, has two types of user account, one being a standard user account with restricted access to many of the administrative files and features of the operating system, and the other a superuser (*root*) account with elevated privileges. Typically, a user can gain root access either by logging in as the root user, or using the *su* - command and entering the root password. In the following example, a user is gaining root access via the *su* - command:

```
[neil@demo-server ~]$ su -  
Password:  
[root@demo-server ~]#
```

Note that the command prompt for a regular user ends with a \$ sign while the root user has a # character. When working with the command-line, this is a useful indication as to whether or not you are currently issuing commands as the root user.

If the *su* - command fails, the root account on the system has most likely been disabled for security reasons. In this case, the *sudo* command can be used instead as outlined below.

Using *sudo*, a single command requiring root privileges may be executed by a non-root user. Consider the following attempt to update the operating system with the latest patches and packages:

```
$ apt update  
Reading package lists... Done  
E: Could not open lock file /var/lib/apt/lists/lock - open (13: Permission denied)
```

Introduction

Optionally, user accounts may be configured so that they have access to root level privileges. Instead of using the `su` - command to first gain root access, user accounts with administration privileges are able to run otherwise restricted commands using `sudo`.

```
$ sudo apt update
[sudo] password for demo:
Hit:1 http://us.archive.ubuntu.com/ubuntu bionic InRelease
.
```

To perform multiple commands without repeatedly using the `sudo` command, a command prompt with persistent super-user privileges may be accessed as follows:

```
[neil@demo-server]$ sudo su -
[neil@demo-server]#
```

The reason for raising this issue so early in the book is that many of the command-line examples outlined in this book will require root privileges. Rather than repetitively preface every command-line example with directions to run the command as root, the command prompt at the start of the line will be used to indicate whether or not the command needs to be performed as root. If the command can be run as a regular user, the command will be prefixed with a `$` command prompt as follows:

```
$ date
```

If, on the other hand, the command requires root privileges, the command will be preceded by a `#` command prompt:

```
# apt install openssh-server
```

1.2 Opening a Terminal Window

If you are running Ubuntu with the GNOME desktop and need to access a command-prompt you will need to open a terminal window. This can be achieved by right-clicking on the desktop background and selecting the Open Terminal menu option as shown in Figure 1-1:

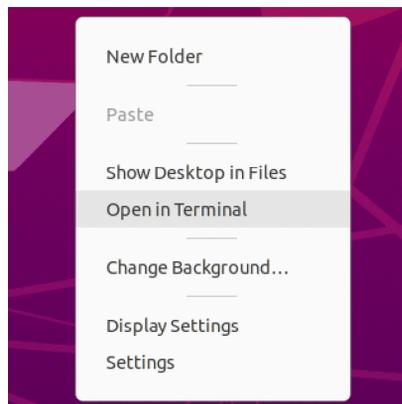


Figure 1-1

A terminal window may also be opened within the GNOME desktop using the Ctrl-Alt-T keyboard accelerator.

1.3 Editing Files

Configuring a Linux system typically involves editing files. For those new to Linux it can be unclear which editor to use. If you are running a terminal session and do not already have a preferred editor we recommend using the *nano* editor. To launch *nano* in a terminal window simply enter the following command:

```
# nano <file>
```

Where <file> is replaced by the path to the file you wish to edit. For example:

```
# nano /etc/passwd
```

Once loaded, *nano* will appear as illustrated in Figure 1-2:

```
GNU nano 2.9.3 /etc/passwd

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nolog$
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,:/run/systemd/netif:/usr$

[ Read 44 lines ]

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^I To Spell  ^_ Go To Line
```

Figure 1-2

To create a new file simply run *nano* as follows:

```
# nano
```

When you have finished editing the file, type Ctrl-S to save the file followed by Ctrl-X to exit. To open an existing file, use the Ctrl-R keyboard shortcut.

If you prefer to use a graphical editor within the GNOME desktop environment *gedit* is a useful starting point for basic editing tasks. To launch *gedit* from the desktop press Alt-F2 to display the Enter a Command window as shown in Figure 1-3:

Introduction

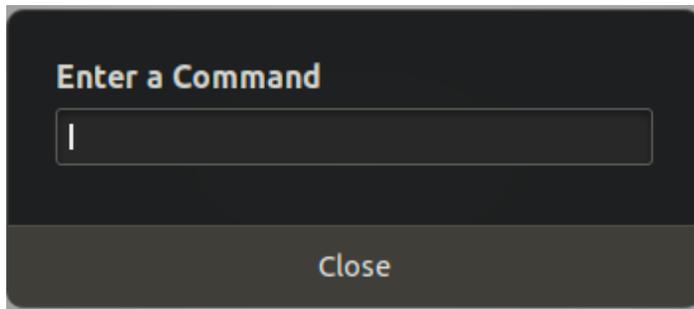


Figure 1-3

Enter *gedit* into the text field and press the Enter key. After a short delay, gedit will load ready to open, create and edit files:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
l1st:x:38:38:Mail1ng L1st Manager:/var/l1st:/usr/sbin/nologin
lrc:x:39:39:lrcd:/var/run/lrcd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106:./home/syslog:/usr/sbin/nologin
messagebus:x:103:107:./nonexistent:/usr/sbin/nologin
_apt:x:104:65534:./nonexistent:/usr/sbin/nologin
uuldd:x:105:111:./run/uuldd:/usr/sbin/nologin
avahi-autopd:x:106:112:Avahi auto1p daemon,,,:/var/lib/avahi-auto1pd:/usr/sbin/nologin
usbmux:x:107:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
dnsmasq:x:108:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
rtkit:x:109:114:RealtimeKit,,,:/proc:/usr/sbin/nologin
cups-pk-helper:x:110:116:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
speech-dispatcher:x:111:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false
whoopsie:x:112:117:./nonexistent:/bin/false
kernoops:x:113:65534:Kernel Oops Tracking Daemon,,,:/usr/sbin/nologin
saned:x:114:119:./var/lib/saned:/usr/sbin/nologin
pulse:x:115:120:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
avahi:x:116:122:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
colord:x:117:123:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
hplip:x:118:7:HPLIP system user,,,:/var/run/hplip:/bin/false
```

Figure 1-4

Alternatively, launch gedit from a terminal window either with or without the path to the file to open:

```
# gedit
# gedit /etc/passwd
```

1.4 Feedback

We want you to be satisfied with your purchase of this book. If you find any errors in the book, or have any comments, questions or concerns please contact us at feedback@ebookfrenzy.com.

1.5 Errata

While we make every effort to ensure the accuracy of the content of this book, it is inevitable that a book covering a subject area of this size and complexity may include some errors and oversights. Any known issues with the book will be outlined, together with solutions, at the following URL:

<https://www.ebookfrenzy.com/errata/ubuntu2004.html>

In the event that you find an error not listed in the errata, please let us know by emailing our support team at *feedback@ebookfrenzy.com*.

2. A Brief History of Linux

Ubuntu Linux is one of a number of variants (also referred to as *distributions*) of the Linux operating system and is the product of a U.K. company named Canonical Ltd. The company was founded in 1994 by Mark Shuttleworth. The origins of Linux, however, go back even further. This chapter will outline the history of both the Linux operating system and Ubuntu.

2.1 What exactly is Linux?

Linux is an operating system in much the same way that Windows is an operating system (and there any similarities between Linux and Windows end). The term operating system is used to describe the software that acts as a layer between the hardware in a computer and the applications that we all run on a daily basis. When programmers write applications, they interface with the operating system to perform such tasks as writing files to the hard disk drive and displaying information on the screen. Without an operating system, every programmer would have to write code to directly access the hardware of the system. In addition, the programmer would have to be able to support every single piece of hardware ever created to be sure the application would work on every possible hardware configuration. Because the operating system handles all of this hardware complexity, application development becomes a much easier task. Linux is just one of a number of different operating systems available today.

2.2 UNIX Origins

To understand the history of Linux, we first have to go back to AT&T Bell Laboratories in the late 1960s. During this time AT&T had discontinued involvement in the development of a new operating system named Multics. Two AT&T engineers, Ken Thompson and Dennis Ritchie, decided to take what they had learned from the Multics project and create a new operating system named UNIX which quickly gained popularity and wide adoption both with corporations and academic institutions.

A variety of proprietary UNIX implementations eventually came to market including those created by IBM (AIX), Hewlett-Packard (HP-UX) and Sun Microsystems (SunOS and Solaris). In addition, a UNIX-like operating system named MINIX was created by Andrew S. Tanenbaum designed for educational use with source code access provided to universities.

2.3 Who Created Linux?

The origins of Linux can be traced back to the work and philosophies of two people. At the heart of the Linux operating system is something called the *kernel*. This is the core set of features necessary for the operating system to function. The kernel manages the system's resources and handles communication between the hardware and the applications. The Linux kernel was developed by Linus Torvalds who, taking a dislike to MS-DOS, and impatient for the availability of MINIX for the new Intel 80386 microprocessor, decided to write his own UNIX-like kernel. When he had

A Brief History of Linux

finished the first version of the kernel, he released it under an open source license that enabled anyone to download the source code and freely use and modify it without having to pay Linus any money.

Around the same time, Richard Stallman at the Free Software Foundation, a strong advocate of free and open source software, was working on an open source operating system of his own. Rather than focusing initially on the kernel, however, Stallman decided to begin by developing open source versions of all the UNIX tools, utilities and compilers necessary to use and maintain an operating system. By the time he had finished developing this infrastructure it seemed like the obvious solution was to combine his work with the kernel Linus had written to create a full operating system. This combination became known as GNU/Linux. Purists insist that Linux always be referred to as GNU/Linux (in fact, at one time, Richard Stallman refused to give press interviews to any publication which failed to refer to Linux as GNU/Linux). This is not unreasonable given that the GNU tools developed by the Free Software Foundation make up a significant and vital part of GNU/Linux. Unfortunately, most people and publications simply refer to Linux as Linux and this will probably always continue to be the case.

2.4 The History of Ubuntu

As mentioned previously, Ubuntu is one of a number of Linux distributions. The source code that makes up the Ubuntu distribution originates from a highly regarded Linux distribution known as Debian created by Ian Murdoch.

A South African internet mogul named Mark Shuttleworth (who made his fortune selling his company to VeriSign for around \$500 million) decided it was time for a more user friendly Linux. He took the Debian distribution and worked to make it a more human friendly distribution which he called Ubuntu. He subsequently formed a company called Canonical Ltd to promote and provide support for Ubuntu.

If you are new to Linux, or already use Linux and want to try a different Linux distribution it is unlikely you will find a better option than Ubuntu.

2.5 What does the word “Ubuntu” Mean?

The word “Ubuntu” is an ancient Zulu and Xhosa word that means “humanity to others”. Ubuntu also means “I am what I am because of who we all are”. It was chosen because these sentiments precisely describe the spirit of the Ubuntu distribution.

2.6 Summary

The origins of the Linux operating system can be traced back to the work of Linus Torvalds and Richard Stallman in the form of the Linux kernel combined with the tools and compilers built by the GNU project.

Over the years, the open source nature of Linux has resulted in the release of a wide range of different Linux distributions. One such distribution is Ubuntu, based on the Debian Linux distribution and created by Canonical Ltd, a company founded by Mark Shuttleworth.

3. Installing Ubuntu on a Clean Disk Drive

There are now three ways in which an Ubuntu system can be deployed. One method is to either purchase new hardware or re-purpose an existing computer system on which to install and run the operating system. Alternatively, a virtualization platform such as VirtualBox or VMware can be used to install and run Ubuntu inside a virtual machine on an existing operating system. Another option is to create a cloud-based operating system instance using services such as Amazon AWS, Google Cloud or Microsoft Azure (to name but a few). Since cloud-based instances are typically created by selecting a pre-configured, ready to run operating system image that is already optimized for the cloud platform, and using that as the basis for the Ubuntu system, there is no need to perform a manual operating system installation in this situation.

If, on the other hand, you plan to install Ubuntu on your own hardware or make use a virtualization environment, the first step on the path to learning about Ubuntu involves installing the operating system.

Ubuntu can be installed either in a clean disk environment (where an entire disk is cleared of any existing partitions and dedicated entirely to Ubuntu) or in a dual boot environment where Ubuntu co-exists with another operating system on the disk (typically a member of the Microsoft Windows family of operating systems).

In this chapter we will be covering the clean disk approach to installation from local or remote installation media. Dual boot installation with a Windows 10 system will be covered in *"Dual Booting Ubuntu with Windows"*.

3.1 Ubuntu Installation Options

Ubuntu can be downloaded free of charge from the following web page:

<https://ubuntu.com/download>

This page provides a number of download options depending on how the operating system is to be installed and used:

- **Ubuntu Desktop** - Downloads the installation media for the desktop edition of the operating system. This edition is intended for use on desktop and laptop systems where a graphical desktop environment is needed and is only available for 64-bit x86 systems. The desktop edition can be downloaded in the form of an ISO image which you can then write to a USB drive using the steps outlined later in this chapter. When booted, the desktop media will allow you to test out Ubuntu by running a Live Ubuntu session prior to performing the installation.

Installing Ubuntu on a Clean Disk Drive

- **Ubuntu Server** - Downloads the installation media for the server edition of the operating system. This image is intended for performing an installation on servers on which the graphical desktop environment is not required and is available for x86, ARM, IBM POWER (PowerPC) and s390x (IBM System z mainframe) systems. The installation media does not include the option to try Ubuntu before installing and uses the text based installer instead of the graphical installer used for Ubuntu Desktop. This allows Ubuntu to be installed on systems without a graphical console.

When downloading Ubuntu Server edition, the following options are available:

- **Standard Live Server ISO Image** - Contains everything to install Ubuntu Server. This allows the installation to be performed without needing a network or internet connection.
- **Network Installer ISO Image** - Contains the minimum needed to begin the installation process during which additional packages are downloaded based on choices made during the configuration phase. The Network installer will be covered in detail in the chapter entitled *“Installing Ubuntu with the Network Installer”*.

Both the Live Server and Network installer images may also be used to perform Preboot Execution Environment (PXE) network installations. When using PXE to install Ubuntu, the Ubuntu image is installed on a specially configured server (referred to as a PXE boot server). The client system on which Ubuntu is to be installed is then configured to boot over the network from the image on the PXE boot server (assuming the client hardware supports PXE) to initiate the installation.

3.2 Server vs. Desktop Editions

Clearly a decision between the Desktop and the Server Edition images needs to be made before installation can begin. If you would like to try Ubuntu before installing it, then the Desktop option is the best solution since it allows you to boot Ubuntu from the installation media without first installing it on a disk drive. As shown in Figure 3-1, this option also allows the installation to be initiated from within the live session:

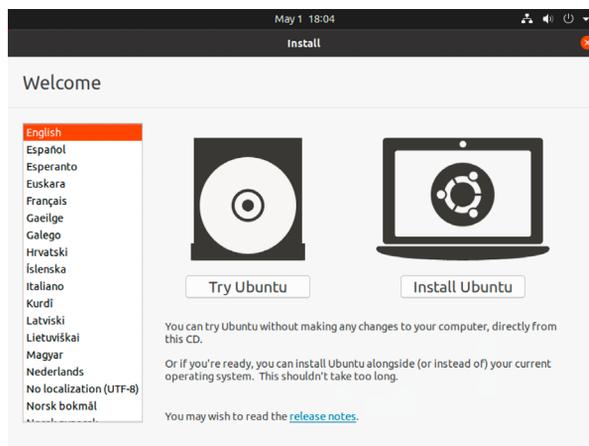


Figure 3-1

If the graphical desktop environment is not required, and the destination system does not have internet access or a graphical console then the Live Server ISO image is recommended since this allows a fully functional server to be built without the need to download any additional packages.

Regardless of the chosen installation method, packages can be added to and removed from the system after installation to configure the system to specific needs.

3.3 Obtaining the Ubuntu Installation Media

For the purposes of this chapter, the Ubuntu Desktop environment will be installed using the graphical installer. Begin, therefore, by downloading the Ubuntu Desktop 20.04 ISO image from the following URL:

<https://ubuntu.com/download/desktop>

The DVD ISO image is self-contained including all of the packages necessary to install an Ubuntu system and is named using the following convention:

```
ubuntu-<version>-<edition>-<architecture>.iso
```

For example, the Ubuntu 20.04 Desktop ISO image for 64-bit Intel/AMD systems is named as follows:

```
ubuntu-20.04-desktop-amd64.iso
```

Having downloaded the image, either burn it to disk or use the steps in the next section to write the media to a USB drive and configure your virtualization environment to treat it as a DVD drive.

3.4 Writing the ISO Installation Image to a USB Drive

These days it is more likely that an operating system installation will be performed from a USB drive than from a DVD. Having downloaded the ISO installation image for Ubuntu, the steps to write that image to a USB drive will differ depending on whether the drive is attached to a Linux, macOS or Windows system. The steps outlined in the remainder of this section assume that the USB drive is new, or has been reformatted to remove any existing data or partitions:

3.4.1 Linux

The first step in writing an ISO image to a USB drive on Linux is to identify the device name. Before inserting the USB drive, identify the storage devices already detected on the system by listing the devices in `/dev` as follows:

```
# ls /dev/sd*  
/dev/sda /dev/sda1 /dev/sda2
```

Attach the USB drive to the Linux system and run the `dmesg` command to get a list of recent system messages, one of which will be a report that the USB drive was detected and will be similar to the following:

```
[445597.988045] sd 6:0:0:0: [sdb] Attached SCSI removable disk
```

This output tells us that we should expect the device name to include “sdb” which we can confirm

Installing Ubuntu on a Clean Disk Drive

by listing device names in */dev* again:

```
# ls /dev/sd*
/dev/sda  /dev/sda1  /dev/sda2  /dev/sdb
```

From this output we can tell that the USB drive has been assigned to */dev/sdb*. The next step before writing the ISO image to the device is to run the *findmnt* command to make sure it has not been auto-mounted:

```
# findmnt /dev/sdb?
TARGET                SOURCE      FSTYPE OPTIONS
/media/demo/C24E-6727 /dev/sdb1  vfat    rw,nosuid,nodev, ...
```

If the *findmnt* command indicates that the USB drive has been mounted, unmount it before continuing:

```
# umount /media/demo/C24E-6727
```

Once the filesystem has been unmounted, use the *dd* command as follows to write the ISO image to the drive:

```
# dd if=/path/to/iso/<image name>.iso of=/dev/sdb bs=512k
```

The writing process can take some time (as long as 10 - 15 minutes) to complete depending on the image size and speed of the system on which it is running. Once the image has been written, output similar to the following will appear and the USB drive is ready to be used to install Ubuntu:

```
4056+1 records in
4056+1 records out
2126544896 bytes (2.1 GB, 2.0 GiB) copied, 625.911 s, 3.4 MB/s
```

3.4.2 macOS

The first step in writing an ISO image to a USB drive attached to a macOS system is to identify the device using the *diskutil* tool. Before attaching the USB device, open a Terminal window and run the following command:

```
$ diskutil list
/dev/disk0 (internal, physical):
#:                                TYPE NAME                SIZE      IDENTIFIER
0:    GUID_partition_scheme      *1.0 TB      disk0
1:    EFI EFI                    209.7 MB    disk0s1
2:    Apple_APFS Container disk2  1000.0 GB   disk0s2

/dev/disk1 (internal):
#:                                TYPE NAME                SIZE      IDENTIFIER
0:    GUID_partition_scheme      28.0 GB     disk1
1:    EFI EFI                    314.6 MB    disk1s1
2:    Apple_APFS Container disk2  27.7 GB     disk1s2

/dev/disk2 (synthesized):
#:                                TYPE NAME                SIZE      IDENTIFIER
```

Installing Ubuntu on a Clean Disk Drive

```
0:      APFS Container Scheme -                    +1.0 TB      disk2
      Physical Stores disk1s2, disk0s2
1:      APFS Volume Macintosh HD                  473.6 GB     disk2s1
2:      APFS Volume Preboot                       42.1 MB      disk2s2
3:      APFS Volume Recovery                      517.0 MB     disk2s3
4:      APFS Volume VM                            1.1 GB       disk2s4
```

Having established a baseline of detected devices, insert the USB drive into a port on the macOS system and run the command again. The same results should appear with one additional entry for the USB drive resembling the following:

```
/dev/disk3 (external, physical):
#:      TYPE NAME                                SIZE         IDENTIFIER
0:      *16.0 GB disk3
```

In the above example, the USB drive has been assigned to `/dev/disk3`. Before proceeding, unmount the disk as follows:

```
$ diskutil unmountDisk /dev/disk3
Unmount of all volumes on disk3 was successful
```

Finally, use the `dd` command to write the ISO image to the device, taking care to reference the raw disk device (`/dev/rdisk3`) and entering your user password when prompted:

```
$ sudo dd if=/path/to/iso/image.iso of=/dev/rdisk3 bs=1m
```

Once the image has been written, the USB drive is ready.

3.4.3 Windows

A number of free tools are available for Windows that will write an ISO image to a USB drive, but one written specifically for writing Linux ISO images is the Fedora Media Writer tool which can be downloaded from the following URL:

<https://getfedora.org/en/workstation/download/>

Once installed, launch the writer tool and select the *Custom image* option as highlighted in Figure 3-2:

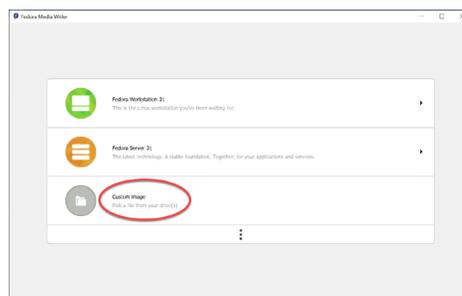


Figure 3-2

In the resulting file selection dialog, navigate to and select the Ubuntu installation ISO image and

Installing Ubuntu on a Clean Disk Drive

click on the *Open* button. After selecting the image, a dialog will appear within which the image can be written to the USB drive. Select the target USB drive from the device menu before clicking on the *Write to Disk* button:

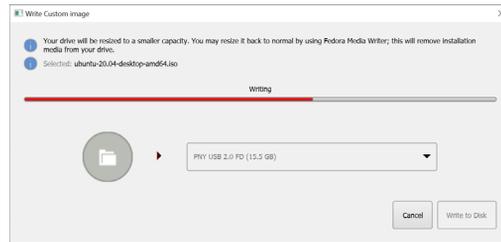


Figure 3-3

Once the image has been written to the device, the device is ready to be used to perform the installation.

3.5 Booting from the Ubuntu USB Image

Insert the Ubuntu installation media into the appropriate drive and power on the system. If the system tries to boot from the hard disk drive you will need to enter the BIOS set up for your computer and change the boot order so that it boots from the installation media drive first. For the first few seconds of the boot process a largely blank screen will appear with the following image located along the bottom edge:



Figure 3-4

If no action is taken at this point, Ubuntu will boot into the Live session and provide the option to either try Ubuntu without installing, or to begin the installation process as shown in Figure 3-1 above. Alternatively, pressing the keyboard Esc key will enter the boot menu system beginning with the language selection screen shown in Figure 3-5:



Figure 3-5

Navigate using the keyboard arrow keys and press the Enter key to select a language at which point the Ubuntu boot menu screen will appear as shown below:



Figure 3-6

Once again, options are provided to either try the Ubuntu Live session, or to begin the installation process. If you experience a black screen when attempting to start or install Ubuntu, reboot the system and try again using one of the *safe graphics* options. The installation media and system memory may also be checked for defects, or the system booted from the primary hard disk drive installed in the computer system (assuming it contains a bootable operating system image). A range of Function key options provide access to settings such as changing the accessibility options, accessing help and specifying special boot parameters.

3.6 Installing Ubuntu

From within either the live session or the boot menu, select the option to begin the Ubuntu installation and wait for the initial screen of the installer to appear:

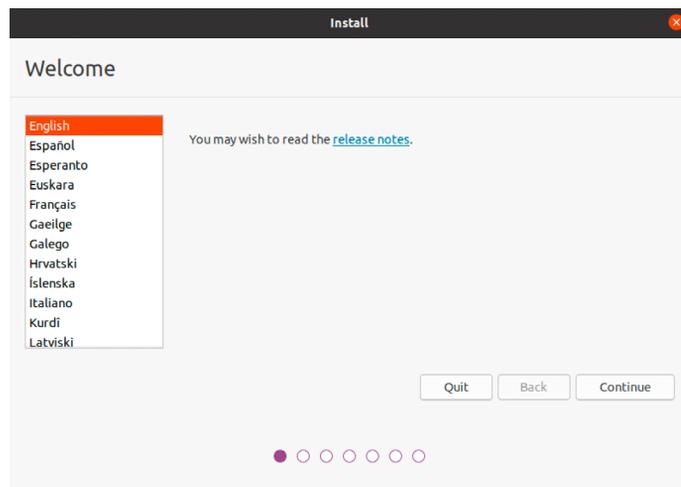


Figure 3-7

Installing Ubuntu on a Clean Disk Drive

On the welcome screen, select your preferred language before clicking on the Continue button to proceed to the next screen:

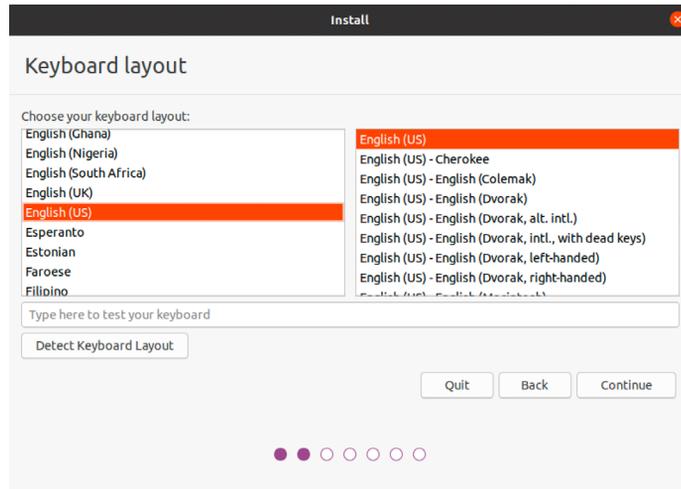


Figure 3-8

Either select your keyboard layout or, if you are unsure, click on the Detect Keyboard Layout button to work through some steps to identify your keyboard before clicking on Continue. On the next screen, choose whether to perform the Normal or Minimal installation:

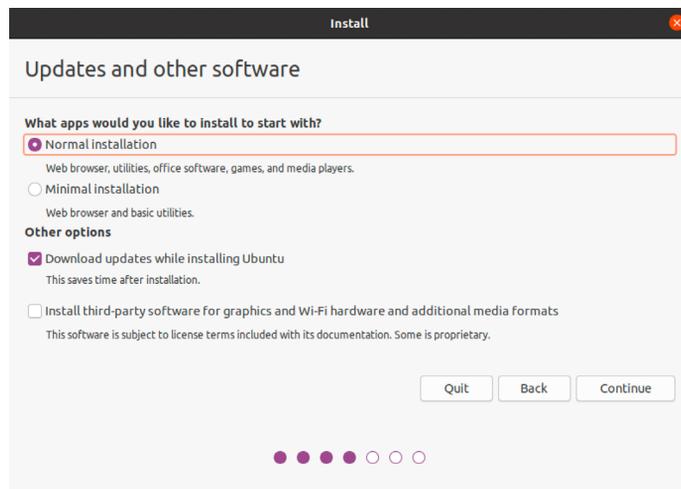


Figure 3-9

Select the Normal option if you have plenty of disk space and want to explore the main applications and utilities included with Ubuntu without having to manually install them later. Alternatively, to avoid cluttering the system with software you may never need, select the Minimal option. Regardless of the choice made here, all of the software provided with Ubuntu can be easily installed or removed at any point in the future if needed.

The option is also available to update the software packages that comprise the Ubuntu distribution during the installation. Ubuntu, as with most actively supported operating systems, continues to be updated with bug fixes and security patches long after it has been released to the public. If this option is selected and the system is connected to the internet, the installer will download any updates issued since the Ubuntu installation image was released and applies them to the system during installation. If you choose not to perform this update during the installation process these updates may still be applied at any time after the installation completes.

A second option provides the choice of whether to install 3rd party non-open source software to support specific hardware devices and the playback of proprietary media files on the system. Some users object fervently to using any software which is not published under a public license. If you are one of those people then do not select this option. If, on the other hand, you just want the best experience from your Ubuntu installation then this option is recommended.

Having made appropriate selections, click the Continue button to proceed to the disk allocation screen:

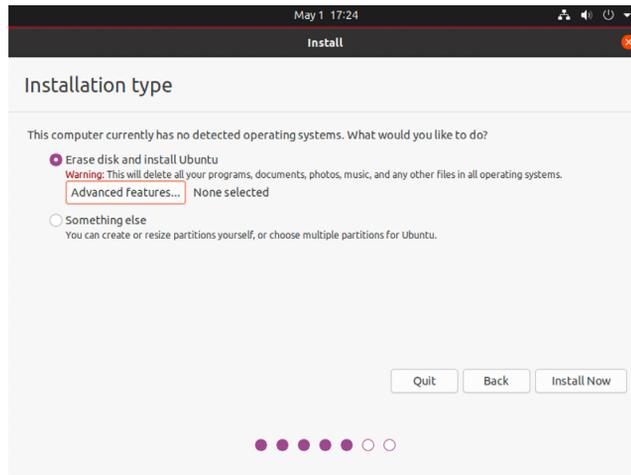


Figure 3-10

Assuming that this is a new disk on which an existing operating system is not present, the installer will provide the option to erase the entire disk and use it for Ubuntu (in which case the installer will calculate and implement a typical and recommended partition layout). Alternatively, to define your own custom disk layout, select the *Something else* option to manually create and size the disk partitions that will contain the operating system and your data.

Clicking on the *Advanced Features* button will provide the option to use Logical Volume Management (LVM). LVM is strongly recommended to make the management of the disks and partitions on the system easier, a topic covered in detail in the chapter entitled “*Adding a New Disk to an Ubuntu Volume Group and Logical Volume*”:

Installing Ubuntu on a Clean Disk Drive

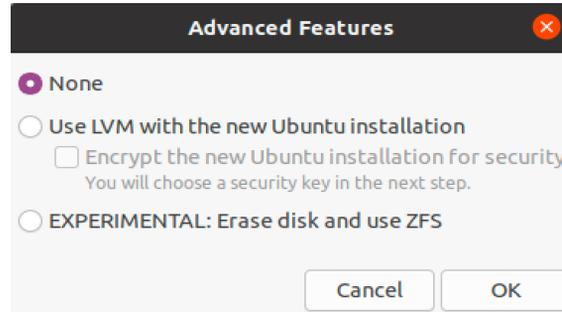


Figure 3-11

If the security of the data stored on the disk is of paramount concern, select the option to encrypt the Ubuntu installation. If this option is selected the next screen will prompt you to choose a security key which will then need to be entered each time the system starts.

The option is also provided to use the ZFS filesystem. ZFS is an advanced filesystem that has many of the features of Logical Volume Management built-in together with a range of other features such as the ability to take disk snapshots and shadowing (copy-on-write). Although ZFS has been around for many years, the implementation on Ubuntu is considered to be experimental at this stage and should not be used for production environments.

Once the selections have been made, click on the Install Now button to begin the installation process at which point the system will seek confirmation that the changes are to be made to the disk drive:

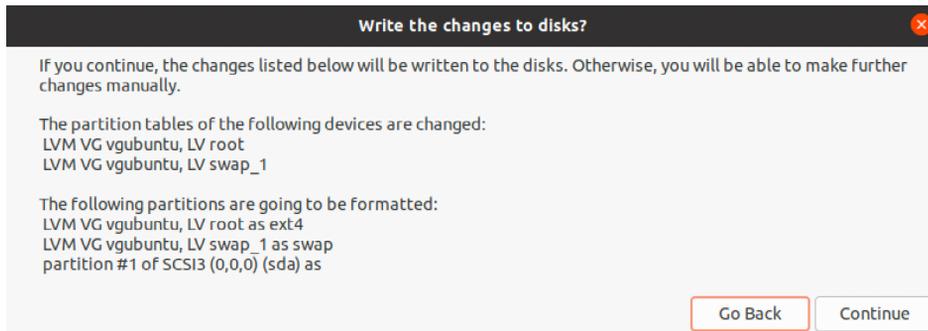


Figure 3-12

While the installation is in progress, the installer will ask for information about your geographical location in order to configure time zone settings:

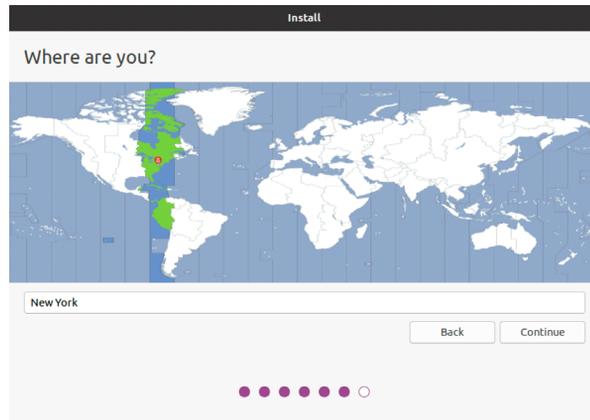


Figure 3-13

Next, the installer will ask you to provide a user name and password for the first account on the system:

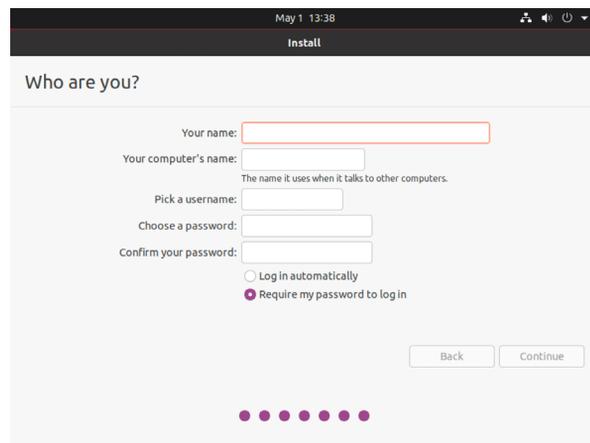


Figure 3-14

The option is also provided to dictate whether the password must be specified each time the user wishes to log into the system. By default, each time Ubuntu starts a login screen will be presented seeking username and password credentials. If you are the sole user of the system and would like to bypass this screen and be logged in automatically each time the system boots, be sure to set the *Log in automatically* checkbox before proceeding.

Once all the questions have been answered, the installer will simply proceed with the installation. Depending on the speed of your system, and whether or not you opted to download updates during the installation, this process can take some time. For a more detailed view of the steps being performed by the installer, click on the status title located above the progress bar:

Installing Ubuntu on a Clean Disk Drive

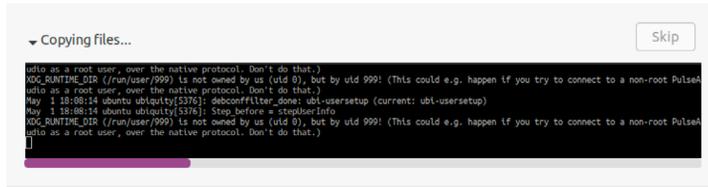


Figure 3-15

When the installation is complete, a dialog will appear to inform you the system is ready to be restarted:

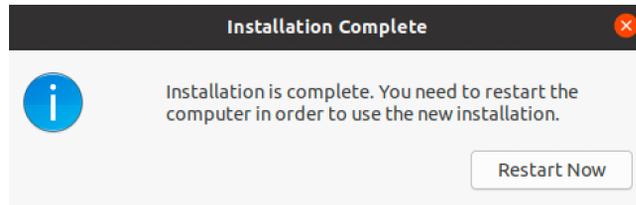


Figure 3-16

When you are ready to reboot, press the Restart Now button. The installer may prompt you to remove the installation media and the system will take a few moments to shut down. At this point remove the USB flash drive and press the Enter key to proceed.

3.7 Accessing the Ubuntu Desktop

Once the system has started, if the password requirement option was enabled the GNOME Display Manager (GDM) login screen (Figure 3-17) will appear. To access the system, select the user name and enter the password specified during installation:

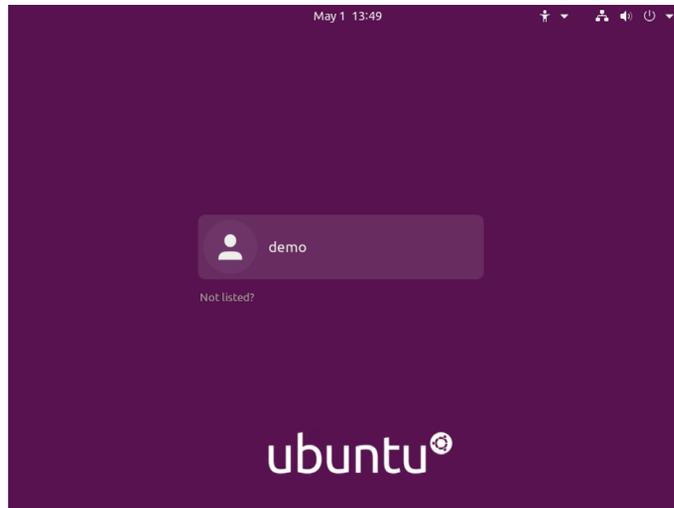


Figure 3-17

Alternatively, if the installation was configured to log directly into the desktop, the GNOME

desktop (Figure 3-18) will appear after the system has restarted:

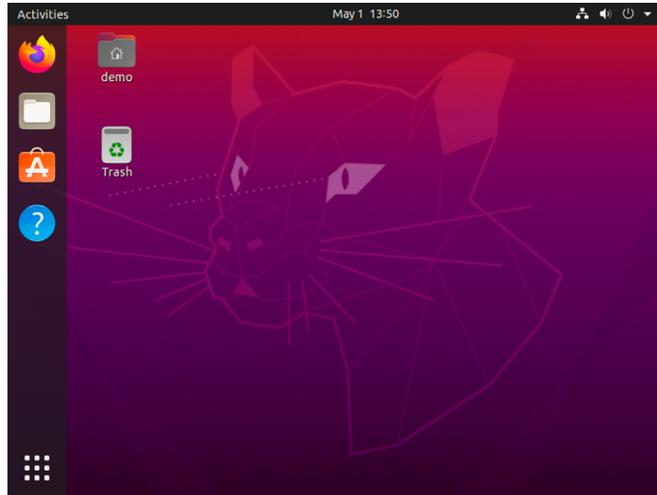


Figure 3-18

3.8 Installing Updates

As with most operating systems today, each particular release of the Ubuntu distribution continues to evolve after it has been released. This generally takes the form of bug fixes and security updates and, occasionally, new features that may be downloaded over the internet and installed on your system.

Best practices dictate that the first step after installing Ubuntu is to make sure any available updates are applied to the system. This can be achieved via the command-line prompt in a Terminal window using the *apt* package manager tool. To check for the availability of updates, right-click on the desktop background and, from the resulting menu, select the Open in Terminal option:

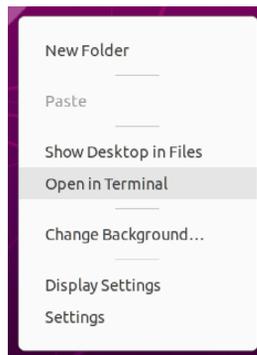


Figure 3-19

Within the Terminal window, run the following commands to gain root privileges and update the package list:

```
$ sudo su -
```

Installing Ubuntu on a Clean Disk Drive

```
# apt update
```

If updated packages are available, the command will display output similar to the following:

```
.  
. .  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
103 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Any pending updates may be applied using the *apt* tool:

```
# apt upgrade
```

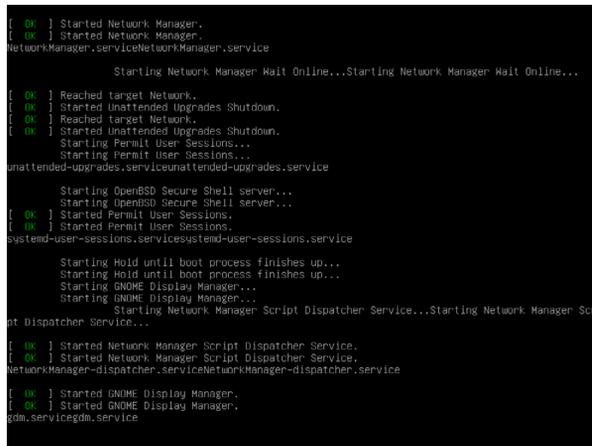
Upon execution, the *apt* tool will provide a list of packages that are available for update and prompt for permission to perform the update

```
103 upgraded, 7 newly installed, 0 to remove and 0 not upgraded.  
Need to get 75.2 MB/286 MB of archives.  
After this operation, 352 MB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

Once the upgrade is complete the installation is essentially finished.

3.9 Displaying Boot Messages

During the boot process, the system will display the Ubuntu splash screen which hides from view all of the boot messages generated by the system as it loads. To make these messages visible during the boot process (as shown in Figure 3-20), simply press the keyboard Esc key while the system is starting:



```
[ OK ] Started Network Manager.  
[ OK ] Started Network Manager.  
NetworkManager.serviceNetworkManager.service  
Starting Network Manager: Wait Online...Starting Network Manager: Wait Online...  
[ OK ] Reached target Network.  
[ OK ] Started Unattended Upgrades Shutdown.  
[ OK ] Reached target Network.  
[ OK ] Started Unattended Upgrades Shutdown.  
Starting Permit User Sessions...  
Starting Permit User Sessions...  
unattended-upgrades.serviceunattended-upgrades.service  
Starting OpenBSD Secure Shell server...  
Starting OpenBSD Secure Shell server...  
[ OK ] Started Permit User Sessions.  
[ OK ] Started Permit User Sessions.  
systemd-user-sessions.service  
Starting Hold until boot process finishes up...  
Starting Hold until boot process finishes up...  
Starting GNOME Display Manager...  
Starting GNOME Display Manager...  
Starting Network Manager Script Dispatcher Service...Starting Network Manager Script Dispatcher Service...  
[ OK ] Started Network Manager Script Dispatcher Service.  
[ OK ] Started Network Manager Script Dispatcher Service.  
NetworkManager-dispatcher.serviceNetworkManager-dispatcher.service  
[ OK ] Started GNOME Display Manager.  
[ OK ] Started GNOME Display Manager.  
gdm.servicegdm.service
```

Figure 3-20

The default behavior can be changed so that messages are always displayed by default by editing the */etc/default/grub* file and changing the GRUB_CMDLINE_LINUX setting which, by default,

will resemble the following:

```
GRUB_CMDLINE_LINUX="... rhgb quiet"
```

If you are new to Linux and are not familiar with the editors available, refer to the editor recommendations outlined in the “*Introduction*” chapter. For example, to use the nano editor, enter the following command to start the editor and load the *grub* file:

```
# nano /etc/default/grub
```

To remove the graphical boot screen so that messages are visible without pressing the Esc key, remove the “splash” and “quiet” options from the setting:

```
GRUB_CMDLINE_LINUX="quiet splash"
```

This change will cause the system to display all of the boot messages generated by the system.

Once the changes have been made, run the following command to generate a new boot configuration to take effect next time the system starts:

```
# grub-mkconfig --output=/boot/grub/grub.cfg
```

3.10 Summary

The first step in working with Ubuntu is to install the operating system. In the case of a cloud-based server, this task is typically performed automatically when an operating system image is selected for the system based on a range of options offered by the cloud service provider. Installation on your own hardware or in a virtual machine, however, involves downloading the installation media in the form of an ISO image, writing that image to suitable storage such as a DVD or USB drive and booting from it. Once installation is complete, it is important to install any operating system updates that may have been released since the original installation image was created.

4. Installing Ubuntu with the Network Installer

The previous chapter explored the different options available when installing Ubuntu with a particular emphasis on using the graphical installer. This chapter will go into more detail regarding the use of the Network installer image to install Ubuntu.

4.1 Network Installer Advantages

The Network installer (also referred to as Netboot) image provides a small image (approximately 50MB in size compared to more than 2GB for the Desktop image) that can be used to install either the server or desktop-based Ubuntu environments. This option requires an internet connection to download additional packages during installation and uses the text-based installer.

While the Server and Desktop images provide very little choice in terms of configuration options during installation, the Network installer provides an extensive list of pre-defined software collections from which to choose during installation. The Ubuntu Desktop ISO image, for example, will only install the GNOME desktop environment. The Network installer, on the other hand, lets you install various desktop environments including KDE Plasma (Kubuntu) and LXQt (Lubuntu). Options are also available to install a basic server environment with optional additional server related packages such as web server, mail server and database software.

All of these packages can, of course, be added or removed after installation is complete when using the Server and Desktop images, but if you want to use a small installation image that can be quickly written to a USB drive and allows you to pre-install many software dependencies, the Network installer is an ideal option.

4.2 Obtaining the Network Installer Image

The Network installer image for Ubuntu 20.04 can be downloaded from the following web page:

<http://archive.ubuntu.com/ubuntu/dists/focal/main/installer-amd64/current/legacy-images/netboot/mini.iso>

Once downloaded, follow the steps in the chapter entitled "Installing Ubuntu on a Clean Disk Drive" to write the *mini.iso* image to a USB drive.

4.3 Booting from the Installer Image

Insert the Ubuntu installation media and power on the system. If the system tries to boot from the hard disk drive you will need to enter the BIOS set up for your computer and change the boot order so that it boots from the installation media drive first. Once the system has booted, the screen shown in Figure 4-1 will appear:

Installing Ubuntu with the Network Installer

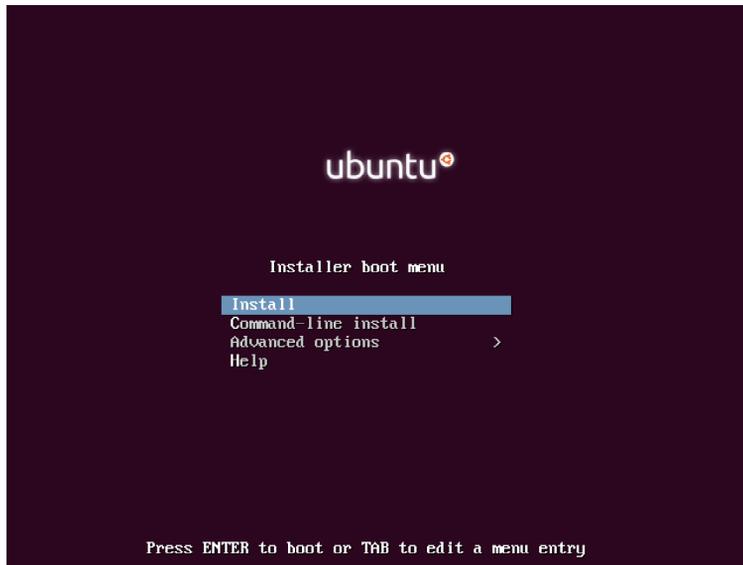


Figure 4-1

4.4 Performing the Installation

From the menu, select the Install option and tap the keyboard Enter key to start the installation process. On the next few screens, use the keyboard arrow keys and the Enter key to select your preferred language and keyboard type.

After making the configuration selections, the installer will scan the device hardware in search of a network interface:

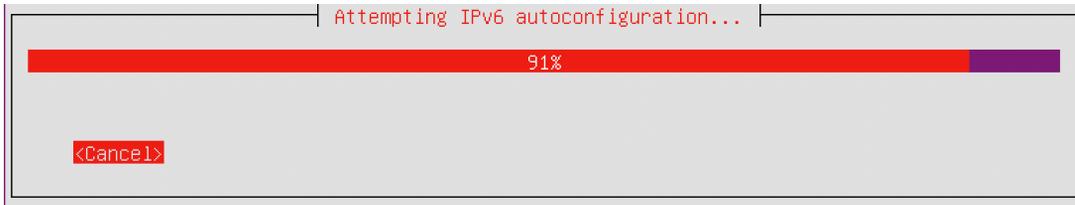


Figure 4-2

Once a network connection has been detected, the system will need to be assigned a host name by which it will appear on the network. When prompted, enter a host name then use the Tab key to navigate to the Continue button before tapping the Enter key to move to the next screen.

During the installation, the installer will download only the packages that are needed to match the software selections you make during the installation process. These Ubuntu software packages are hosted on mirror servers throughout the world. To ensure optimal download performance select the country in which the system is located:

Installing Ubuntu with the Network Installer

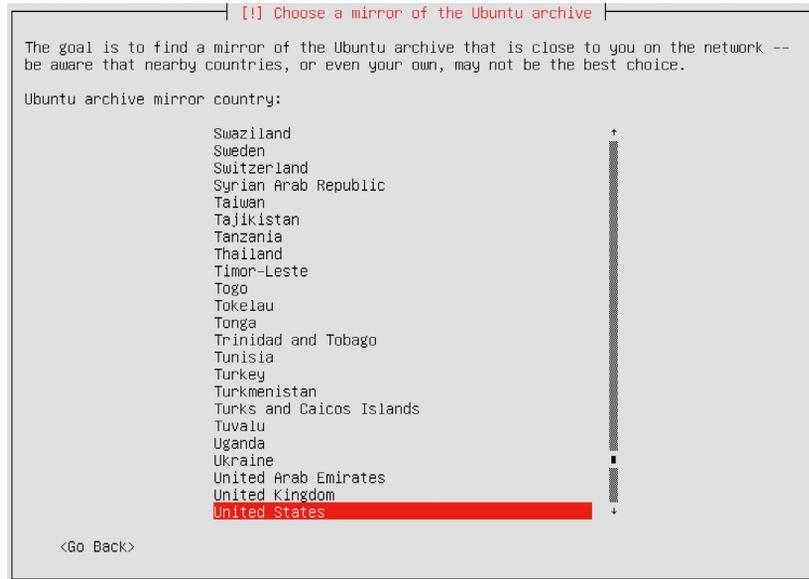


Figure 4-3

On the next screen select a mirror server before continuing:



Figure 4-4

If the computer on which the installation is being performed is located behind a proxy based firewall, enter the information for the proxy on the screen shown in Figure 4-5. A proxy service firewall is placed between the internet and an internal network of computers and acts as a go-between for the two environments. With a proxy service in place, internal client computers do not connect directly to outside resources. Instead they connect to the proxy server which in turn connects with the external resource on behalf of the client, thereby masking the internal IP address of the client. Any responses from the external resources are handled by the proxy service which passes them along to the client that originally requested the data:

Installing Ubuntu with the Network Installer

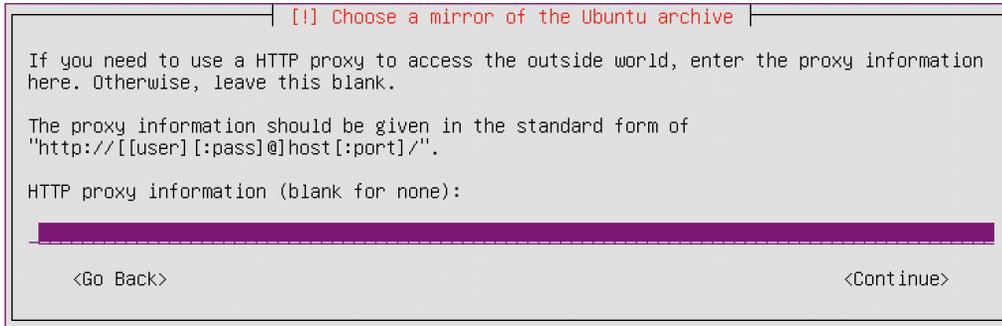


Figure 4-5

If proxy information is not required, simply leave this field blank and proceed to the next screen.

At this point, the installer will download some basic packages needed to start the installation. Once complete, a series of screens will appear asking for the full name, username and password for the first user to be added to the system. Once this information has been provided, the installer will identify the geographical location of the system based on the external IP address of the internet connection. If the installer has detected the location correctly, accept the location, otherwise make the selection manually.

4.5 Disk Partitioning

The next phase of the installation setup involves deciding how the disk drive is to be partitioned and managed:

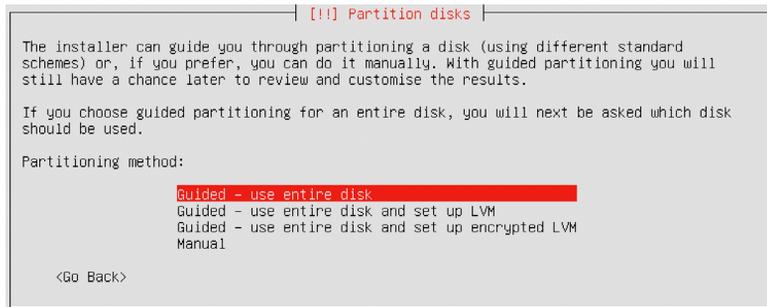


Figure 4-6

Assuming that this is a new disk on which an existing operating system is not present, the installer will provide the option to erase the entire disk and use it for Ubuntu (in which case the installer will calculate and implement a typical and recommended partition layout). If the security of the data stored on the disk is of paramount concern, select the option to encrypt the Ubuntu installation. If this option is selected a later screen will prompt you to choose a security key which will then need to be entered each time the system starts.

The option to use Logical Volume Management (LVM) is also strongly recommended to make the management of the disks and partitions on the system easier, a topic covered in detail in the

chapter entitled “*Adding a New Disk to an Ubuntu Volume Group and Logical Volume*”.

Alternatively, to define your own custom disk layout or to implement more advanced configurations such as RAID, select the Manual option:

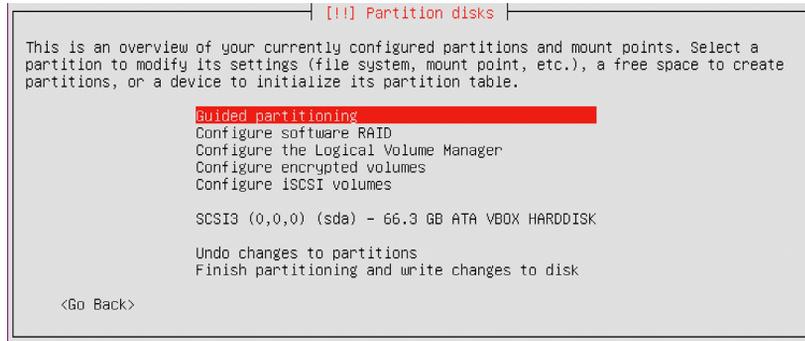


Figure 4-7

Unless you have a specific requirement, select the option to perform a guided LVM installation and follow the steps to confirm the target disk drive and to select the amount of available space to be allocated to the partition. Finally, review the partition configuration summary and select the *Finish partitioning and write changes to disk* menu option as shown in Figure 4-8:

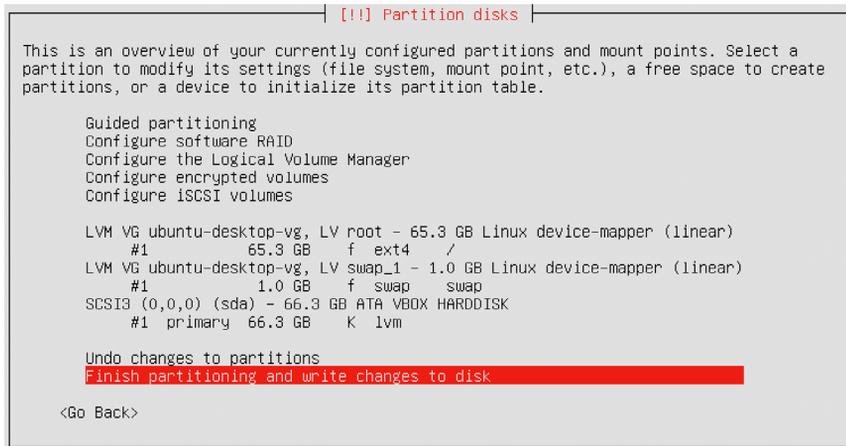


Figure 4-8

Provide a final confirmation and then wait while the partition is created and the base system installed.



Figure 4-9

Installing Ubuntu with the Network Installer

On the next screen, select your preferred option for installing updates on the running system. It is generally recommended to have updates installed automatically, though the manual option is also available if you prefer to control when updates are installed. If, on the other hand, you are managing multiple Ubuntu systems using the Landscape web management environment, select the Landscape option from the menu:

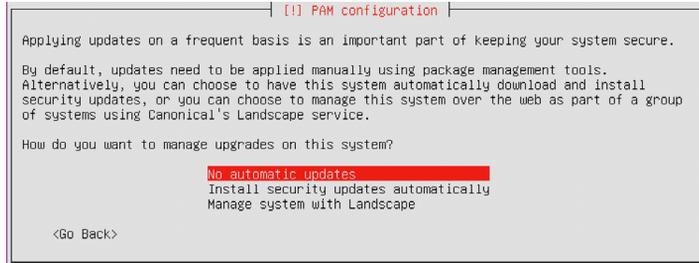


Figure 4-10

4.6 Software Collection Selection

The final step in the network installation is to choose the software collections to be installed. This is performed in the screen shown in Figure 4-11:

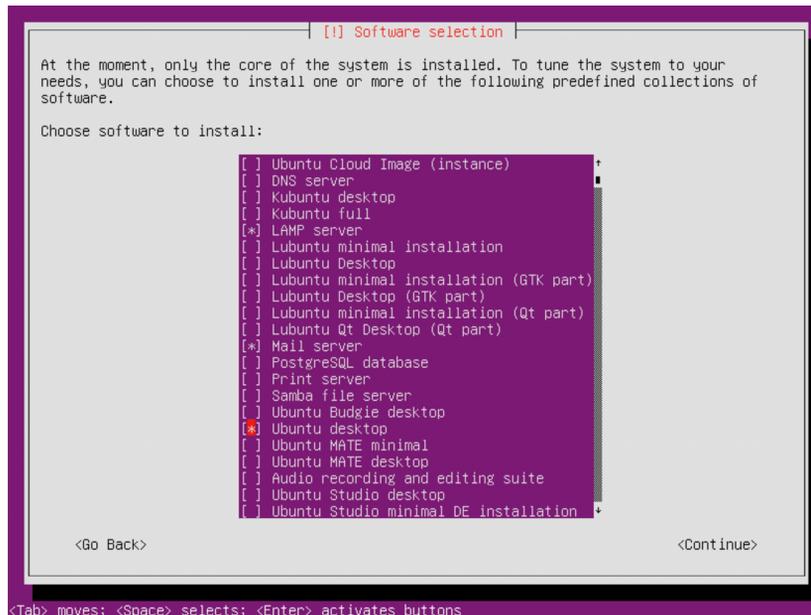


Figure 4-11

Based on your requirements, select the collections you believe you will need (keeping in mind that packages can be added and removed later from within the running system).

Once you have placed asterisks next to all of the items you need, select the Continue option and wait while the selected packages are downloaded and installed. Once the software has been installed, a

screen will appear seeking confirmation that the master boot record on the disk can be modified to add the Ubuntu boot options. Finally, the installer will seek confirmation that the system is to use Coordinated Universal Time (UTC). Accept the use of UTC unless the system is dual booting with Windows, in which case this option should not be selected.

Once the installation is complete, remove the installation media, press the Enter key and wait while the system reboots.

4.7 Installing Software Collections After System Setup

Once the system is up and running, any of the software collections listed in the dialog in Figure 4-11 above can be installed manually from within a running Ubuntu system using the *tasksel* tool. This tool can be installed in a terminal window using the following command:

```
# apt install tasksel
```

Once installed, run the command as follows:

```
# tasksel
```

Once *tasksel* starts, the screen shown in Figure 4-12 will appear from which collections may be selected and installed:

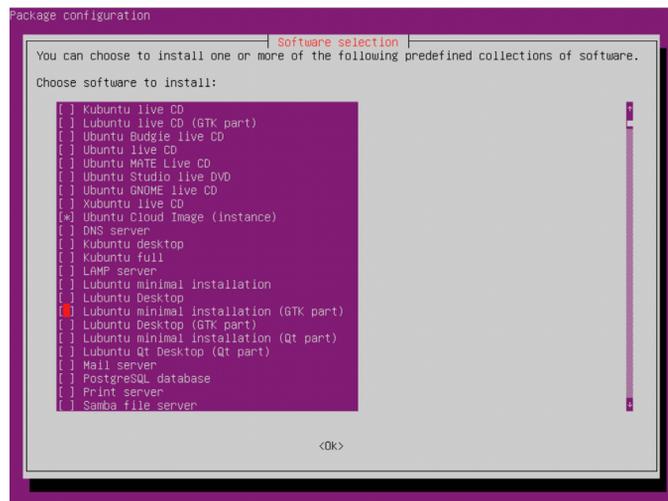


Figure 4-12

4.8 Summary

The Ubuntu Network installation provides a way to install Ubuntu using a relatively small ISO image. This requires that the target system has an active network connection that will be used to install the software packages during the installation process, and limits the installation to being performed in text mode (though the graphical desktop can still be installed). Unlike the desktop and server installation images, the Network installer also allows different software collections to be selected for installation, providing greater control over the installation process and resulting in a more finely configured operating system.

5. Dual Booting Ubuntu with Windows

Ubuntu, just like most Linux distributions, will happily co-exist on a hard disk drive with just about any version of Windows. This is a concept known as dual-booting. Essentially, when you power up your PC you will be presented with a menu providing the option to boot either Ubuntu or Windows. Obviously you can only run one operating system at a time, but it is worth noting that the files on the Windows partition of your disk drive will be available to you from Ubuntu regardless of whether your Windows partition was formatted using NTFS, FAT16 or FAT32.

During the installation process the Ubuntu installer will detect the Windows installation on the drive and provide the option of deleting it and using the entire disk for Ubuntu, or sharing the disk with Windows. In the latter case you will be able to specify the percentage of the disk to be used for Ubuntu.

According to the Ubuntu documentation, a minimum of 25 GB of disk space is required for a full Ubuntu Desktop Edition installation, and more space to store any files you may subsequently create. Once the disk space for Ubuntu has been selected, the installer will resize the Windows partition (without destroying any data) and use the remainder of the disk for Ubuntu.

This chapter will demonstrate how to set up a dual boot system with Ubuntu and Windows 10, change the system that boots by default when the computer system is powered on, and outline how to access files located on the Windows partition of the disk from within Ubuntu.

Within this chapter the assumption is made that the steps outlined in the previous chapter have been followed to create a bootable Ubuntu installation USB drive using the Ubuntu Desktop installation image, and that the system has been configured to boot from the USB drive when started.

5.1 Beginning the Ubuntu Installation

To start the installation, insert the Ubuntu USB drive and reboot your computer system. If the system loads Windows again you will need to change the boot order in your system settings. Details on how to do this will be system specific and were covered briefly in the previous chapter.

Once Ubuntu has loaded you will be presented with the screen shown in Figure 5-1.

Dual Booting Ubuntu with Windows

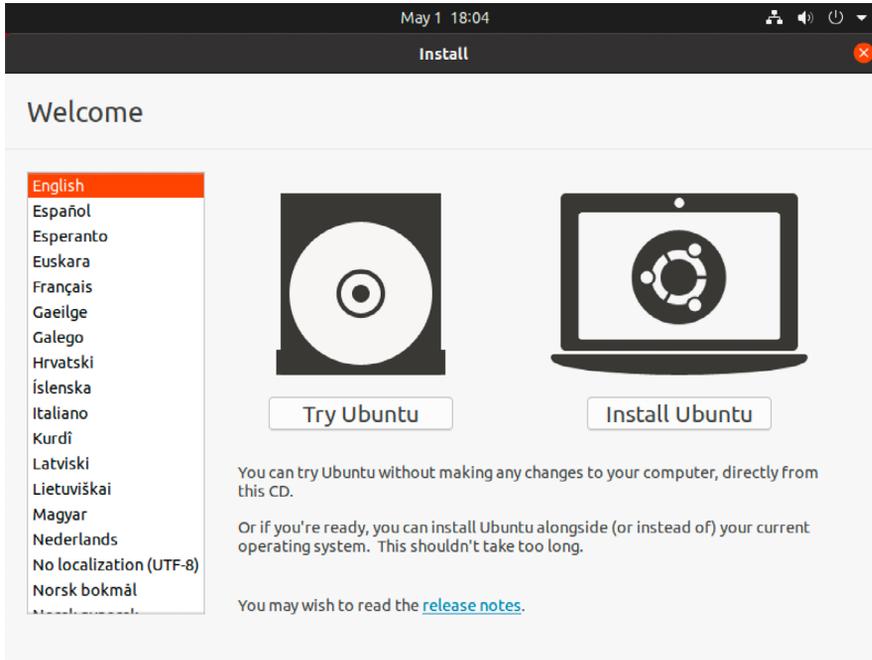


Figure 5-1

Begin the installation by clicking on the Install Ubuntu button and wait for the initial screen of the installer to appear as shown in Figure 5-2:

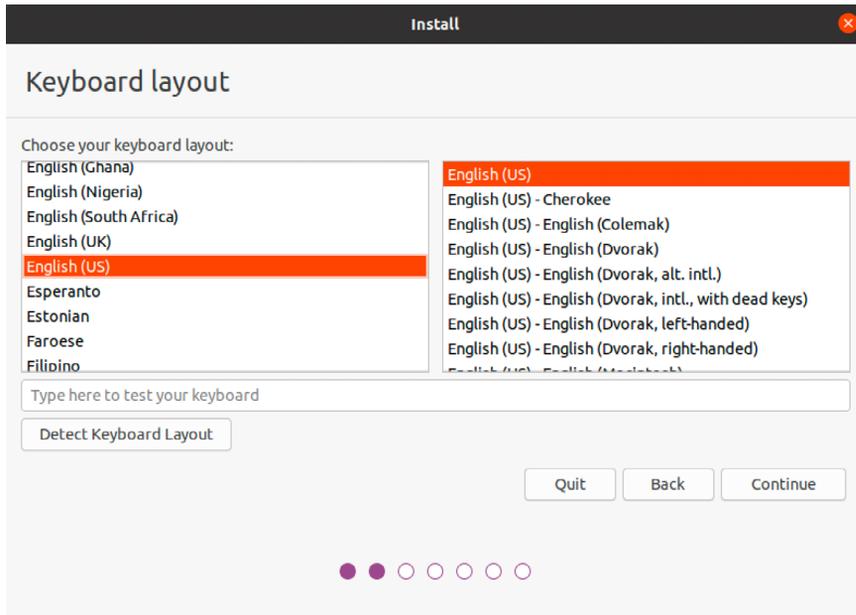


Figure 5-2

Either select your keyboard layout or, if you are unsure, click on the Detect Keyboard Layout button to work through some steps to identify your keyboard before clicking on Continue. On the next screen, choose whether to perform the Normal or Minimal installation:

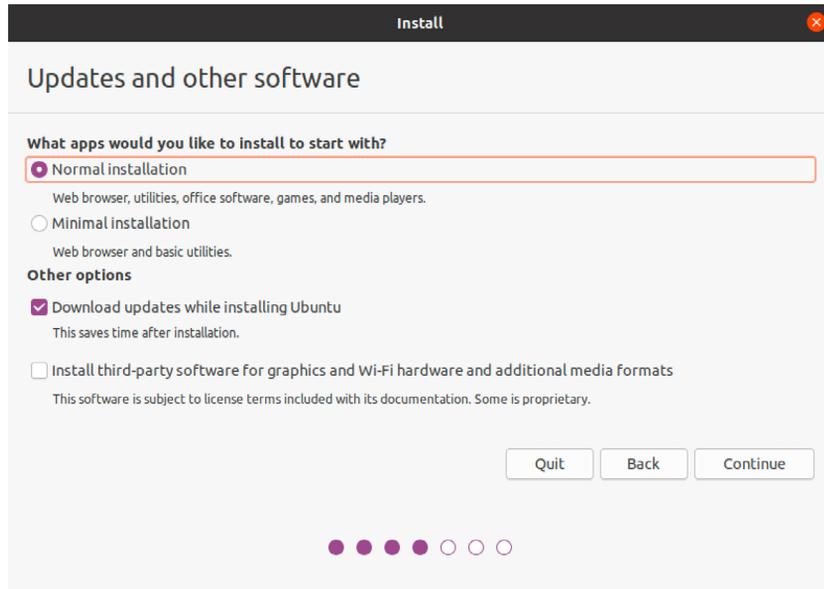


Figure 5-3

Select the Normal option if you have plenty of disk space and want to explore the main applications and utilities included with Ubuntu without having to manually install them later, otherwise use the Minimal option.

The option is also available to update the software packages that comprise the Ubuntu distribution during the installation. If this option is selected and the system is connected to the internet, the installer will download any updates issued since the Ubuntu installation image was released and applies them to the system during installation. If you choose not to perform this update during the installation process these updates may still be applied at any time after the installation completes.

A second option provides the choice of whether to install 3rd party non-open source software to support specific hardware devices and the playback of proprietary media files on the system. Unless you prefer to avoid using non-open source software, use of this option is recommended.

Having made appropriate selections, click the Continue button to proceed to the disk allocation screen. At this point, the installer will have detected the presence of an existing Windows operating system on the target disk drive and will provide a number of options in terms of how the disk should be used to accommodate the Ubuntu installation:

Dual Booting Ubuntu with Windows

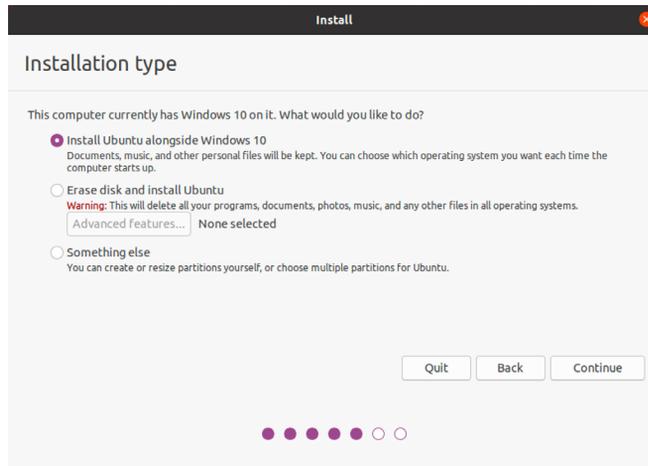


Figure 5-4

Given that it is our intention to configure a dual boot system, the option to Install Ubuntu alongside Windows is the desired choice. With this option selected, click the Continue button to proceed. The subsequent screen allows the amount of disk space allocated to each operating system (in this case Windows and Ubuntu) to be configured:

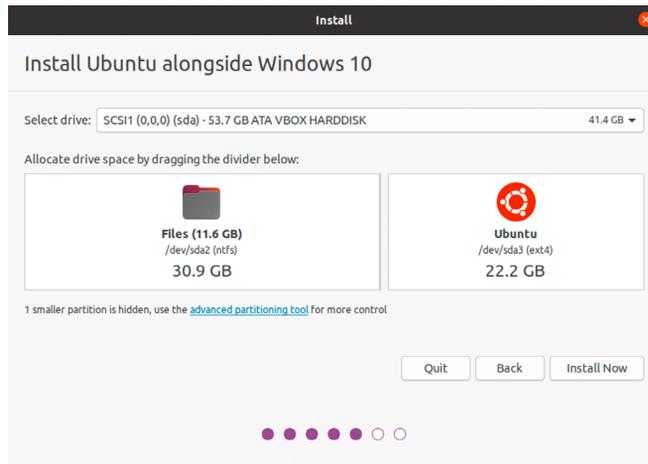


Figure 5-5

At this point it is necessary to decide how much of your Windows partition you wish to donate to the Ubuntu installation. Move the slider (positioned between the Windows and Ubuntu partitions in the diagram) until the Ubuntu partition allocation is a size you are comfortable with. At least 25GB of space should be made available for a full Ubuntu Desktop installation.

When you have selected the size of the Ubuntu partition, click the Install Now button to proceed with the installation. When the warning dialog appears read it carefully and click Continue if you are sure you wish to proceed.

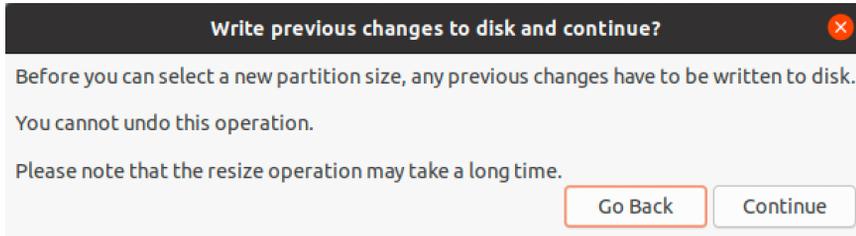


Figure 5-6

While the installation is in progress, the installer will ask for information about your geographical location in order to configure time zone settings:

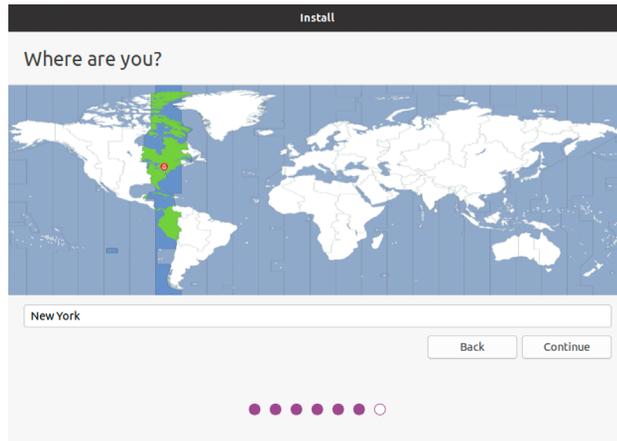


Figure 5-7

Next, the installer will ask you to provide a user name and password for the first account on the system:

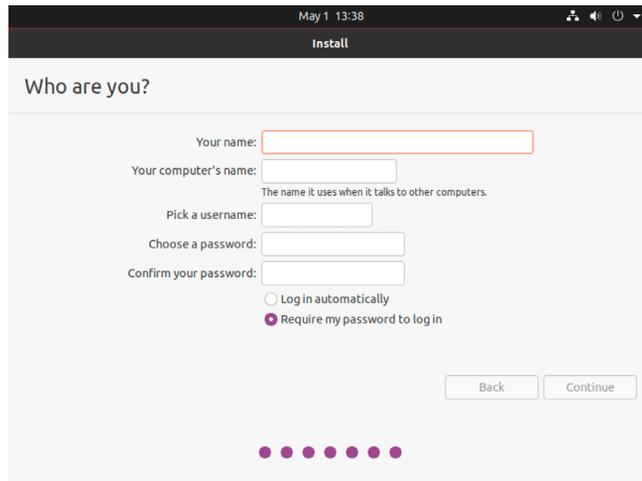


Figure 5-8

Dual Booting Ubuntu with Windows

The option is also provided to dictate whether the password must be specified each time the user wishes to log into the system. By default, each time Ubuntu starts a login screen will be presented seeking username and password credentials. If you are the sole user of the system and would like to bypass this screen and be logged in automatically each time the system boots, be sure to set the Log in automatically checkbox before proceeding.

Once all the questions have been answered, the installer will simply proceed with the installation. Depending on the speed of your system, and whether or not you opted to download updates during the installation, this process can take some time. For a more detailed view of the steps being performed by the installer, click on the status title located above the progress bar:

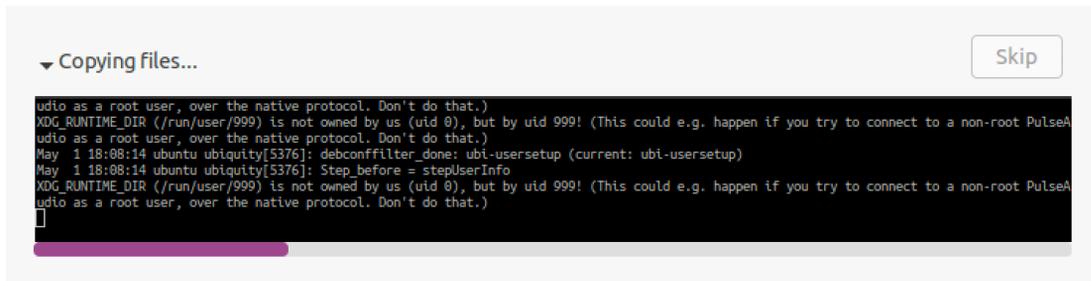


Figure 5-9

When the installation is complete, a dialog will appear to inform you the system is ready to be restarted:

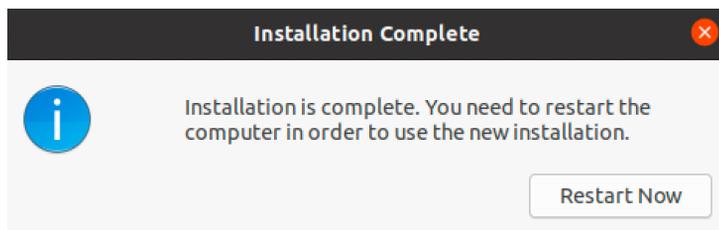


Figure 5-10

When you are ready to reboot, press the Restart Now button. The installer may prompt you to remove the installation media and the system will take a few moments to shut down. At this point remove the USB flash drive and press the Enter key to proceed.

5.2 Booting Ubuntu for the First Time

When the system reboots a screen similar to the one illustrated below will appear providing the option to boot either Windows or Ubuntu:

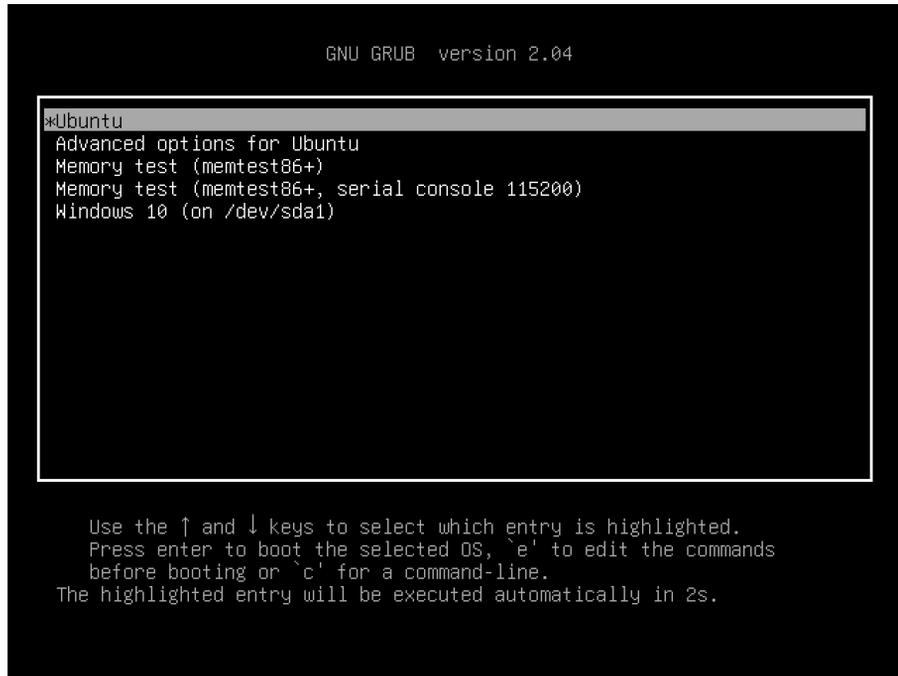


Figure 5-11

Press Enter to boot Ubuntu. If you wish to boot Windows use the keyboard arrow keys to select the Windows option. If you choose to boot Ubuntu the operating system will load and the Ubuntu login screen will appear (unless the automatic login option was selected). Enter the user name and password you set up during the installation process and you will be logged into the Ubuntu Desktop environment.

5.3 Changing the Default Boot Option

When the system starts, the boot options screen will appear and wait 10 seconds for the user to make an operating system choice. If no selection has been made before the timeout elapses, the default operating system will be started. On a newly configured system, the default operating system will be the standard Ubuntu image. This default can, however, be changed from within Ubuntu.

A range of boot configuration options (including the 10 second timeout and the boot settings outlined in “*Installing Ubuntu on a Clean Disk Drive*”) are declared in the `/etc/default/grub` file which reads as follows on a new installation:

```
GRUB_DEFAULT=0
GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""
```

Dual Booting Ubuntu with Windows

The first step in changing the default boot system is to declare the `GRUB_SAVEDDEFAULT` setting within this file and to change the `GRUB_DEFAULT` setting to *saved*:

```
GRUB_DEFAULT=saved
GRUB_SAVEDDEFAULT=true
GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=10
.
.
```

This setting allows a new default value to be saved within the boot configuration. Next, run the *grub-set-default* command to change the default setting using a numbering system that counts the first option as 0. For example, if the Windows 10 option is position 5 in the menu, the command to make Windows 10 the default boot option would read as follows:

```
# grub-set-default 4
```

Check that the new setting has taken effect by running the following command:

```
# grub-editenv list
saved_entry=4
```

Note that the *saved_entry* value is now set to 4. After changing the default, regenerate the boot configuration file as follows:

```
# grub-mkconfig --output=/boot/grub/grub.cfg
```

Reboot the system and verify that the boot menu defaults to the Windows 10 option and that Windows loads after the timeout expires.

5.4 Accessing the Windows Partition from the Command-line

When running Ubuntu in a dual boot configuration it is still possible to access files located on the Windows partition. This can be achieved by manually mounting the partition from the command-line or from the desktop using the Disks (*gnome-disks*) graphical utility.

When working from the command-line, the first step in this process is to create a directory to use as the mount point for our Windows partition. In this example we will create a directory named */mnt/windows*:

```
# mkdir /mnt/windows
```

In order to identify the device name that has been assigned to the Windows partition, use the *fdisk* command as follows:

```
# fdisk -l
.
.
Disk /dev/sda: 50 GiB, 53687091200 bytes, 104857600 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
```

```
Disk identifier: 0x7ef44412
```

```
Device      Boot      Start          End      Sectors   Size Id Type
/dev/sda1   *          2048        1187839   1185792   579M  7 HPFS/NTFS/exFAT
/dev/sda2           1187840    59770533  58582694   28G  7 HPFS/NTFS/exFAT
/dev/sda3           59770878  104855551  45084674  21.5G  5 Extended
/dev/sda5           59770880  104855551  45084672  21.5G  83 Linux
.
.
```

In the above output, the main Windows partitions can be identified from the NTFS entry in the type column. In the above example output the system contains one physical disk drive referenced by device name `/dev/sda`. The first is the Windows system partition while the second, much larger, NTFS partition is the Windows boot partition containing the Windows operating system and user data. This partition contains the files we need access to and is represented by `/dev/sda2`.

With this knowledge, we need to run the `mount` command (assuming the Windows partition is `/dev/sda2`) as follows:

```
# mount /dev/sda2 /mnt/windows
```

Check that the mount was successful by listing the contents of the top level directory of the mount point:

```
# ls /mnt/windows
`$Recycle.Bin`          ProgramData            swapfile.sys
`Documents and Settings` `Program Files`        `System Volume Information`
pagefile.sys           `Program Files (x86)`  Users
PerfLogs               Recovery               Windows
```

To automate the mount each time the system is booted, simply add the appropriate mount line to the `/etc/fstab` file:

```
/dev/sda2 /mnt/windows ntfs defaults 0 0
```

To unmount the Windows file system at any time:

```
# umount /mnt/windows
```

5.5 Accessing the Windows Partition from the Desktop

The first step in mounting a partition using the desktop environment is to launch the disks utility. With the desktop loaded, right-click on the desktop background and select Open Terminal from the resulting menu. Within the terminal window, gain root privileges before launching the Disks utility as follows:

```
$ gnome-disks
```

Once the disks tool has launched, a window similar to that shown in Figure 5-12 will appear:

Dual Booting Ubuntu with Windows

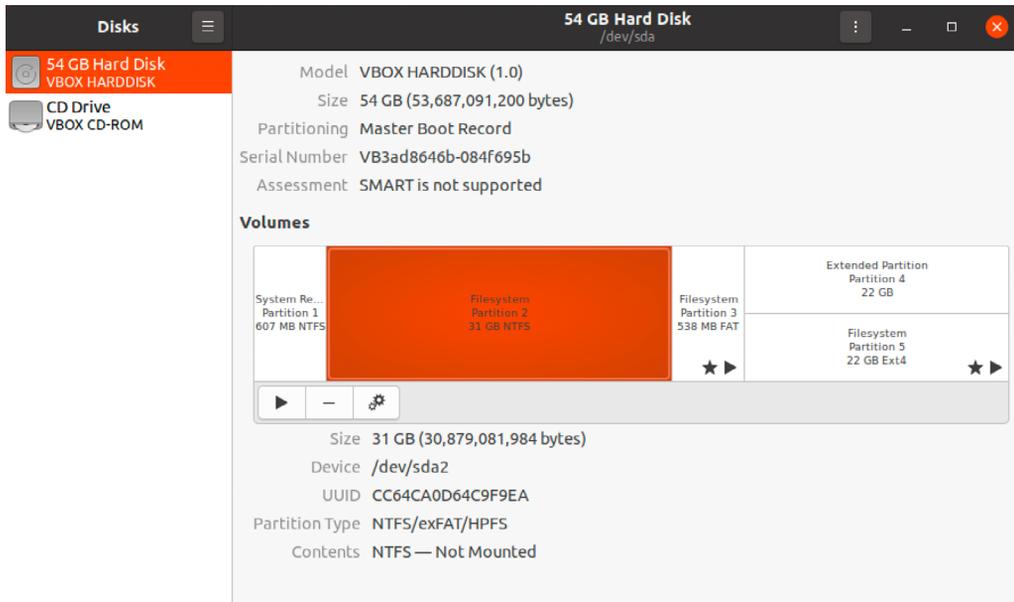


Figure 5-12

To view information about a partition, select it from the graphical representation in the Volumes section of the dialog. In the above example, the Windows system partition is selected and highlighted. To mount the partition so that the contents can be accessed from within Ubuntu, select the partition and click on the options button indicated in Figure 5-13 to display the menu:

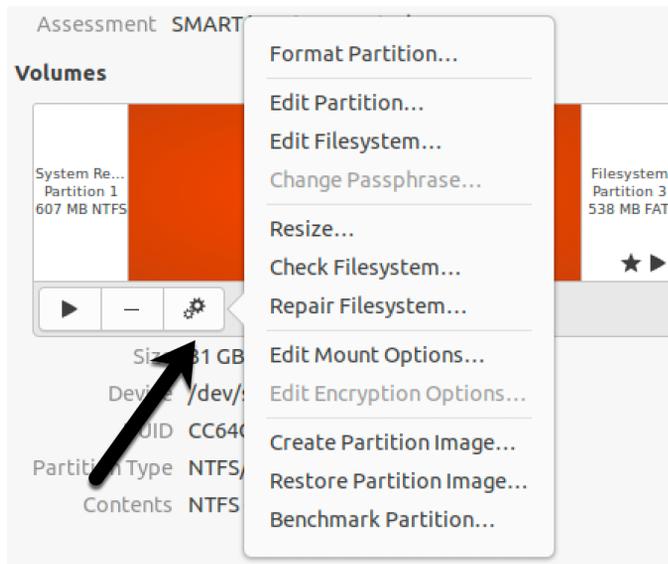


Figure 5-13

From the menu, select *Edit Mount Options...* to display the dialog shown in Figure 5-14. Turn off

the User Session Defaults switch and make sure that both the *Mount at system startup* and *Show in user interface* options are enabled. Within the Mount Point text field, change the path to */mnt/windows*. If the partition needs to be automatically mounted each time the system reboots, also enable the *Mount at system startup* option:

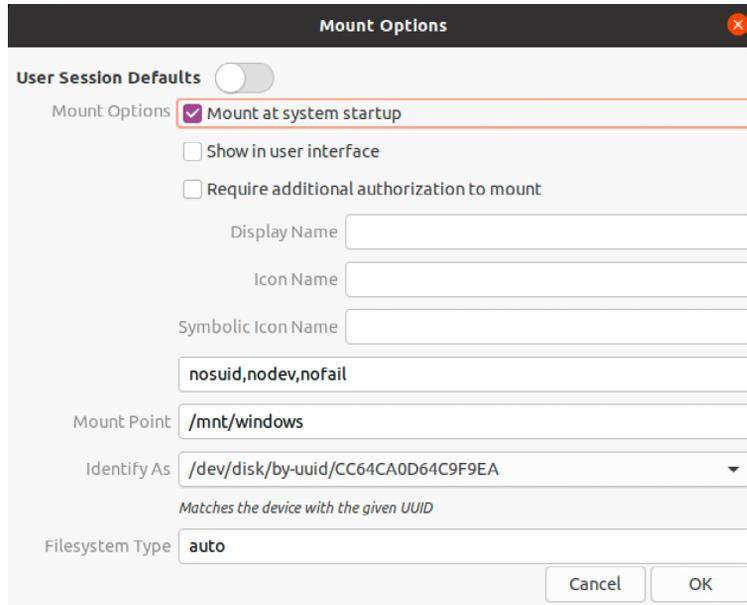


Figure 5-14

With the changes made, click on the OK button to return to the main dialog. To mount the partition, simply click on the mount button highlighted in Figure 5-15 below:

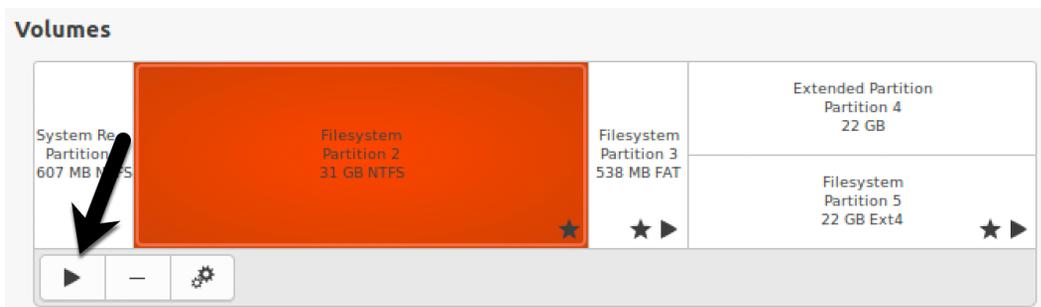


Figure 5-15

Once the partition is mounted, exit from *gnome-disks* and note that an icon for the volume has appeared in the dock as shown on Figure 5-16 (note that if the *Show in user interface* option has been disabled this icon would not be present):

Dual Booting Ubuntu with Windows



Figure 5-16

Double click on the icon to browse the partition using the Files tool:

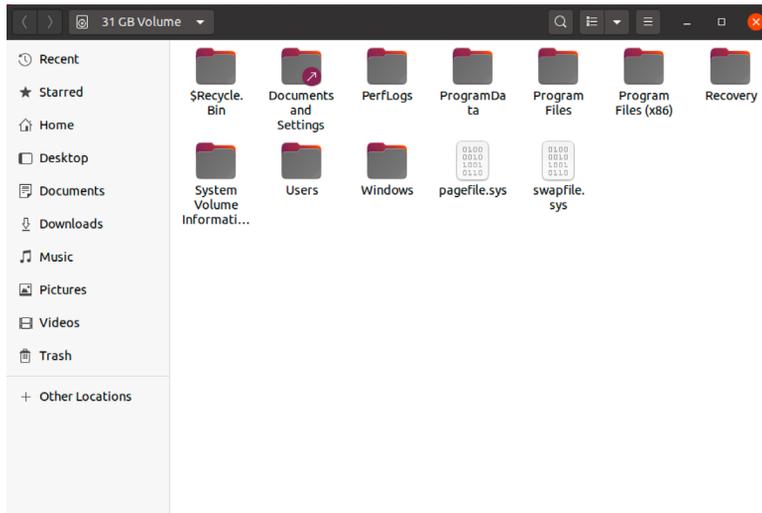


Figure 5-17

To unmount the partition, click on the mount button (which will have changed to a stop button) as shown in Figure 5-15 above.

5.6 Summary

Ubuntu can safely co-exist on the same disk drive as a Windows operating system by creating a dual boot environment. This involves shrinking the amount of space occupied by the Windows system to make room for Ubuntu during the installation. Once Ubuntu has been installed, the boot menu configuration can be modified to change the default operating system. The Windows partition on the disk drive may be accessed from Ubuntu either via the command-line or using the Disks desktop utility.

6. Allocating Windows Disk Partitions to Ubuntu

In the previous chapter we looked at how to install Ubuntu on the same disk as Windows. This so called “dual boot” configuration allows the user to have both operating systems installed on a single disk drive with the option to boot one or the other when the system is powered on.

This chapter is intended for users who have decided they like Ubuntu enough to delete Windows entirely from the disk, and use the resulting space for Linux. In the following sections we will work through this process step by step.

6.1 Unmounting the Windows Partition

If the steps in the “*Dual Booting Ubuntu with Windows*” chapter were followed to mount the Windows partition from within Ubuntu, steps should be taken to unmount the partition before continuing with this chapter. Assuming that the Windows partition was mounted as `/mnt/windows`, it can be unmounted as follows:

```
# umount /mnt/windows
```

The `/etc/fstab` file should also be edited to remove the `/mnt/windows` auto-mount if it was previously added.

6.2 Deleting the Windows Partitions from the Disk

The first step in freeing up the Windows partition for use by Ubuntu is to delete that partition. Before doing so, however, it is imperative that any data you need to keep is backed up from both the Windows and Ubuntu partitions. Having done that, it is safe to proceed with this chapter.

In order to remove the Windows partitions we first need to identify the disk on which they reside using the `fdisk` tool:

```
# fdisk -l
Disk /dev/loop0: 3.7 MiB, 3862528 bytes, 7544 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
.
.
Device      Boot    Start        End    Sectors   Size Id Type
/dev/sda1   *            2048    1187839  1185792   579M  7 HPFS/NTFS/exFAT
/dev/sda2                1187840  59770533  58582694   28G  7 HPFS/NTFS/exFAT
/dev/sda3                59770878 104855551  45084674  21.5G  5 Extended
/dev/sda5                59770880 104855551  45084672  21.5G  83 Linux
```

Allocating Windows Disk Partitions to Ubuntu

In the above example output the system contains one physical disk drive referenced by device name `/dev/sda`. On that disk drive are five partitions accessed via the device names `/dev/sda1` through `/dev/sda5` respectively. Based on the values in the System column, there are two NTFS partitions. The first is the Windows system partition while the second, much larger, NTFS partition is the Windows boot partition containing the Windows operating system and user data. On some systems, an additional Windows Recovery partition may be listed and may also be deleted to free up space for Ubuntu. If such a partition exists, it will likely be listed as follows::

```
/dev/sda3          967012352 976771071    9758720    4.7G 27 Hidden NTFS WinRE
```

To remove the partitions, start the `fdisk` tool using the device name of the disk containing the partition (`/dev/sda` in this instance) and follow the instructions to once again display the partition and sector information:

```
# fdisk /dev/sda
```

```
Welcome to fdisk (util-linux 2.31.1).
```

```
Changes will remain in memory only, until you decide to write them.
```

```
Be careful before using the write command.
```

```
Command (m for help): p
```

```
Disk /dev/sda: 50 GiB, 53687091200 bytes, 104857600 sectors
```

```
Units: sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disklabel type: dos
```

```
Disk identifier: 0x7ef44412
```

Device	Boot	Start	End	Sectors	Size	Id	Type
<code>/dev/sda1</code>	*	2048	1187839	1185792	579M	7	HPFS/NTFS/exFAT
<code>/dev/sda2</code>		1187840	59770533	58582694	28G	7	HPFS/NTFS/exFAT
<code>/dev/sda3</code>		59770878	104855551	45084674	21.5G	5	Extended
<code>/dev/sda5</code>		59770880	104855551	45084672	21.5G	83	Linux

```
Command (m for help):
```

Currently, the Windows system partition is listed as being the bootable partition. Since we will be deleting this partition, the Linux boot partition needs to be marked as bootable. In the above configuration, this is represented by `/dev/sda3`. Remaining within the `fdisk` tool, make this the bootable partition as follows:

```
Command (m for help): a
```

```
Partition number (1,3-5, default 5): 3
```

```
The bootable flag on partition 3 is enabled now.
```

Before proceeding, make a note of the start and end addresses of the partitions we will be deleting (in other words the start of `/dev/sda1` and the sector before the start of `/dev/sda3`).

At the command prompt, delete the Windows partitions (these being partitions 1 and 2 on our example system):

```
Command (m for help): d
Partition number (1-5, default 5): 1
```

Partition 1 has been deleted.

```
Command (m for help): d
Partition number (2-5, default 5): 2
```

Partition 2 has been deleted.

Now that we have deleted the Windows partitions we need to create the new partition in the vacated disk space. The partition number must match the number of the partition removed (in this case 1) and is going to be a primary partition. It will also be necessary to enter the Start and End sectors of the partition exactly as reported for the old partition (*fdisk* will typically offer the correct values by default, though it is wise to double check). If you are prompted to remove the NTFS signature, enter Y:

```
Command (m for help): n
Partition type
   p   primary (0 primary, 1 extended, 3 free)
   l   logical (numbered from 5)
Select (default p): p
Partition number (1,2,4, default 1): 1
First sector (2048-104857599, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-59770877, default 59770877):
```

```
Created a new partition 1 of type 'Linux' and of size 28.5 GiB.
Partition #1 contains a ntfs signature.
```

```
Do you want to remove the signature? [Y]es/[N]o: Y
```

The signature will be removed by a write command.

Having made these changes the next step is to check that the settings are correct (taking this opportunity to double check that the Linux boot partition is bootable):

```
Command (m for help): p
Disk /dev/sda: 50 GiB, 53687091200 bytes, 104857600 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
```

Allocating Windows Disk Partitions to Ubuntu

Disk identifier: 0x7ef44412

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sda1		2048	59770877	59768830	28.5G	83	Linux
/dev/sda3	*	59770878	104855551	45084674	21.5G	5	Extended
/dev/sda5		59770880	104855551	45084672	21.5G	83	Linux

Filesystem/RAID signature on partition 1 will be wiped.

To commit the changes we now need to write the new partition information to disk and quit from the *fdisk* tool:

```
Command (m for help): w
The partition table has been altered.
Syncing disks.
```

6.3 Formatting the Unallocated Disk Partition

In order to make the new partition suitable for use by Ubuntu, it needs to have a file system created on it. The recommended file system type for the current release of Ubuntu is XFS which will be covered in greater detail in the chapter entitled “*Adding a New Disk Drive to an Ubuntu System*”. Creation of the file system is performed using the *mkfs.xfs* command as follows (installing the *xfsprogs* package if necessary):

```
# apt install xfsprogs
# mkfs.xfs /dev/sda1
meta-data=/dev/sda1          isize=512    agcount=4, agsize=1867776 blks
=                           sectsz=512   attr=2, projid32bit=1
=                           crc=1       finobt=1, sparse=0, rmapbt=0,
reflink=0
data      =                  bsize=4096  blocks=7471103, imaxpct=25
=                           sunit=0     swidth=0 blks
naming   =version 2         bsize=4096  ascii-ci=0 ftype=1
log      =internal log     bsize=4096  blocks=3647, version=2
=                           sectsz=512   sunit=0 blks, lazy-count=1
realtime =none             extsz=4096  blocks=0, rtextents=0
```

6.4 Mounting the New Partition

Next, we need to mount the new partition. In this example we will mount it in a directory named */data*. You are free, however, to mount the new partition using any valid mount point you desire or to use it as part of a logical volume (details of which are covered in the chapter entitled “*Adding a New Disk to an Ubuntu Volume Group and Logical Volume*”). First we need to create the directory to act as the mount point:

```
# mkdir /data
```

Secondly, we need to edit the mount table in */etc/fstab* so that the partition is automatically mounted each time the system starts. At the bottom of the */etc/fstab* file, add the following line to mount the new partition (modifying the */dev/sda1* device to match your environment):

```
/dev/sda1 /data xfs defaults 0 0
```

Finally, we can manually mount the new partition (note that on subsequent reboots this will not be necessary as the partition will automount as a result of the setting we added to the */etc/fstab* file above).

```
# mount /data
```

To check the partition, run the following command to display the available space:

```
# df -h /data
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       29G   62M   29G   1% /data
```

6.5 Editing the Boot Menu

The next step is to modify the Ubuntu boot menu. Since this was originally a dual boot system, the menu is configured to provide the option of booting either Windows or Ubuntu. Now that the Windows partition is gone, we need to remove this boot option. On Ubuntu this can be achieved by running the *update-grub* command as follows:

```
# update-grub
Sourcing file `/etc/default/grub'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.3.0-42-generic
Found initrd image: /boot/initrd.img-5.3.0-42-generic
Found linux image: /boot/vmlinuz-5.3.0-28-generic
Found initrd image: /boot/initrd.img-5.3.0-28-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
```

Since there is now only one operating system to boot, the system will automatically boot Ubuntu on the next restart without displaying the boot menu. If you need to access this boot menu, for example to use the advanced boot options, simply press the Esc key during the initial stages of the startup process.

6.6 Using the GNOME Disks Utility

The *gnome-disks* utility provides a user-friendly graphical alternative to reclaiming the Windows partitions from the command-line. Since the example used here will convert the Windows NTFS partitions to XFS format, the first step is to install the *xfsprogs* package as follows:

```
# apt install xfsprogs
```

Once the package has been installed, open a terminal window and launch the *gnome-disks* utility:

```
# gnome-disks
```

After a short delay, the *gnome-disks* tool will appear as shown in Figure 6-1:

Allocating Windows Disk Partitions to Ubuntu



Figure 6-1

In the above example, the disk contains two Windows NTFS partitions which will need to be deleted. Any NTFS partitions with a star shown in the disk map (as is the case for the highlighted partition in the above figure) will need to be unmounted before they can be deleted. This can be achieved by selecting the partition and clicking on the unmount button as indicated in Figure 6-2 below:

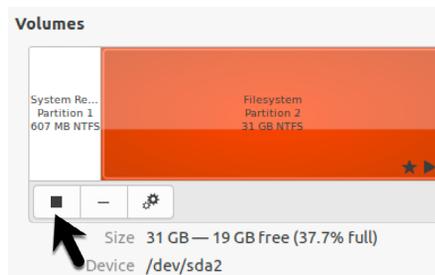


Figure 6-2

With all the NTFS partitions unmounted, the next step is to delete them. Select the left-most partition in the disk map and click on the Delete button as shown in Figure 6-3:

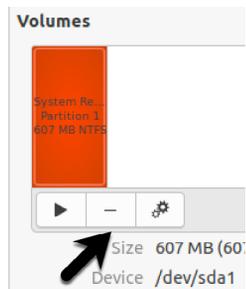


Figure 6-3

Review the information in the confirmation dialog before clicking on the Delete button to commit the change. Once the first partition has been deleted, repeat this step for any remaining NTFS

partitions.

Once the NTFS partitions have been removed, the space should now be shown as being free within the disk map.

A new partition now needs to be created to make use of this free space. With the space selected, click on the new partition button (indicated by the arrow in Figure 6-4):



Figure 6-4

In the Create Partition dialog, choose whether the partition is to occupy all of the available space, or reduce the size if you plan to use the space for more than one partition:

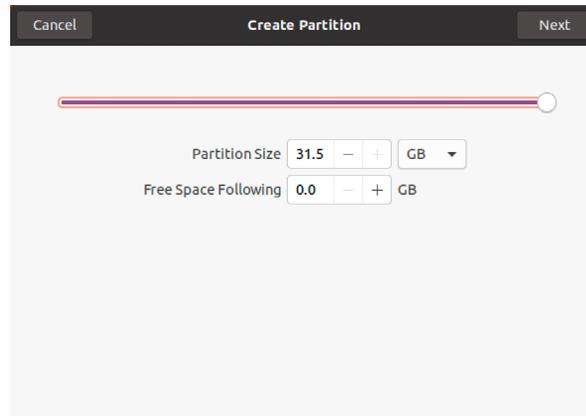


Figure 6-5

Click next and, on the subsequent screen, enter a name for the new partition and choose whether the partition should be erased during creation. This causes the creation process to take more time but is a recommended option. Finally, choose a filesystem type for the partition. This will typically be either ext4 or XFS. Given the benefits of XFS, select Other before clicking the Next button:

Allocating Windows Disk Partitions to Ubuntu

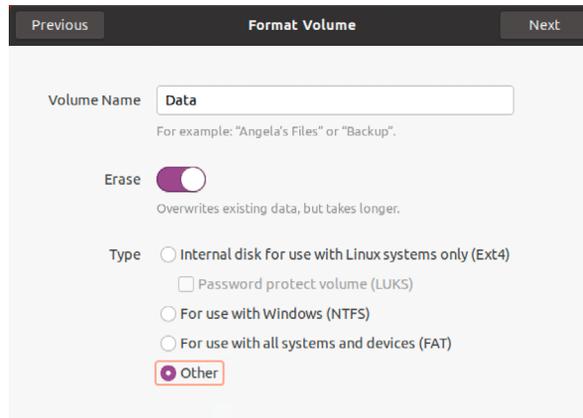


Figure 6-6

On the next screen, select the XFS option before clicking on the Create button:

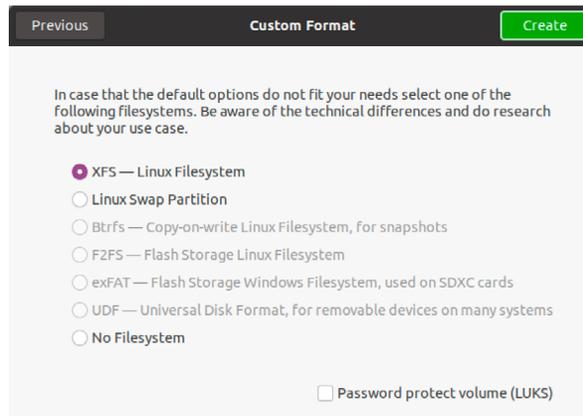


Figure 6-7

If the XFS option is disabled, exit from `gnome-disks` and install the `xfspgros` package before trying again:

```
# apt install xfspgros
```

The `gnome-disks` utility will begin the formatting process and display the status of the process:

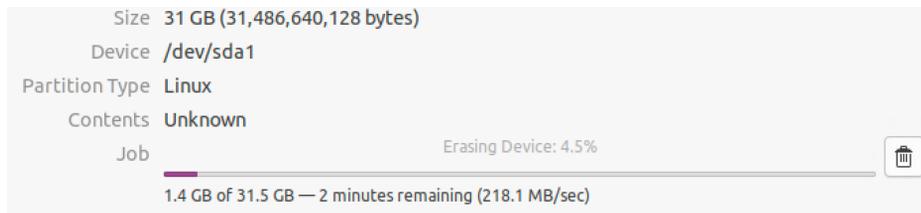


Figure 6-8

Once the partition is ready, it can be mounted either from the command-line or using the `gnome-`

disks utility. To configure a mount point, select the partition and click on the settings button as shown in Figure 6-9:

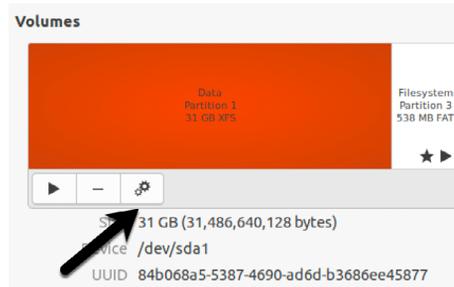


Figure 6-9

From the settings menu, select the *Edit Mount Options...* item to display the dialog shown in Figure 6-10.

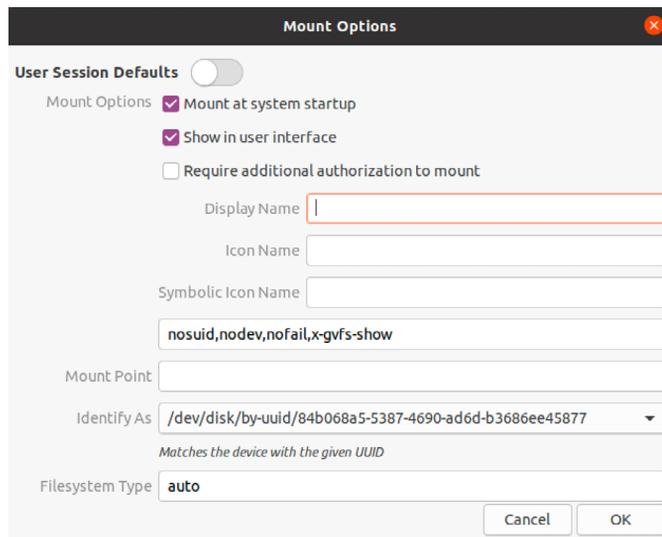


Figure 6-10

Turn off the User Session Defaults switch and configure the mount point to your requirements. In the above figure, the partition is mounted at */data* at system startup and is configured to be identified by the label “Data”.

Once the settings are complete, click on OK. The volume is now mounted and ready for use:

```
$ df -h /mnt/Data
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       29G   62M   29G   1% /data
```

Finally, update the boot menu to remove the Windows option using the steps outlined in section

Allocating Windows Disk Partitions to Ubuntu

6.5 above.

6.7 Summary

The Windows partitions in a dual boot configuration can be removed at any time to free up space for an Ubuntu system by identifying which partitions belong to Windows and then deleting them. Once deleted, the unallocated space can be used to create a new filesystem and mounted to make it available to the Ubuntu system. The final task is to remove the Windows option from the boot menu configuration. These tasks can be performed either from the command-line using *fdisk*, or from within the desktop environment using *gnome-disks*.

7. A Guided Tour of the GNOME 3 Desktop

Ubuntu 20.04 includes the GNOME 3 desktop environment. Although lacking the complexity of Windows and macOS desktops, GNOME 3 provides an uncluttered and intuitive desktop environment that provides all of the essential features of a windowing environment with the added advantage that it can be learned quickly.

In this chapter, the main features of the GNOME desktop will be covered together with an outline of how basic tasks are performed.

7.1 Installing the GNOME Desktop

If the Ubuntu Desktop image was used for the Ubuntu 20.04 installation process, the GNOME desktop will already be installed and will automatically launch each time the system starts.

If any other software configuration was selected during the Ubuntu 20.04 installation process, the GNOME desktop will not have been included in the packages installed on the system. On server-based systems without a display attached, the idea of installing a graphical desktop environment may seem redundant. It is worth noting, however, that remote access to the GNOME desktop is also possible so, even on so called *headless servers* (i.e. servers lacking a monitor, keyboard and mouse) it may still be beneficial to install the GNOME desktop packages. The topic of establishing remote desktop access will be covered in detail in the “*Ubuntu Remote Desktop Access with Vino*” chapter of this book.

If the installation configuration did not include the GNOME desktop, it may be installed at any time using the following command:

```
# apt install tasksel  
# tasksel install ubuntu-desktop
```

Once the installation is complete, the desktop login screen will appear next time the system restarts.

7.2 An Overview of the GNOME 3 Desktop

The screen shown in Figure 7-1 below shows the appearance of a typical, newly launched GNOME desktop session before any other programs have been launched or configuration changes made:

A Guided Tour of the GNOME 3 Desktop

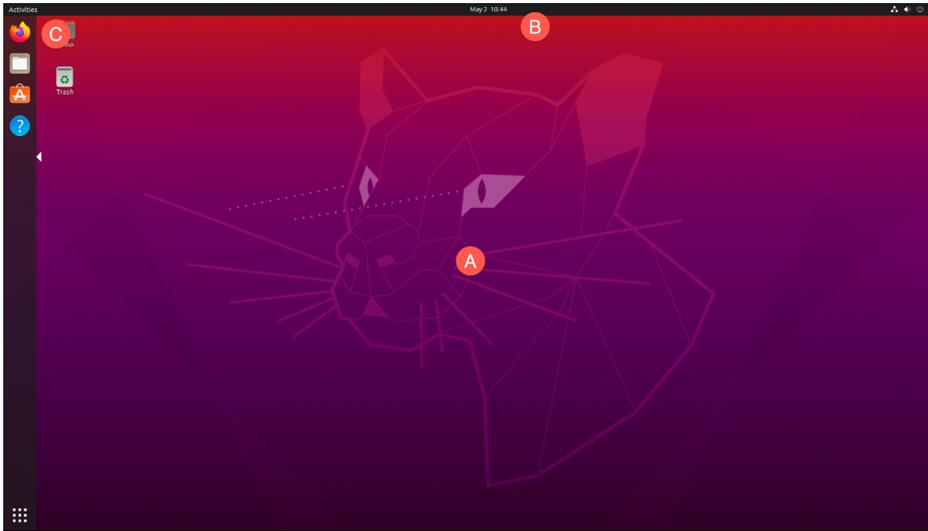


Figure 7-1

The main desktop area (marked A) is where windows will appear when applications and utilities are launched.

The bar at the top of the screen (B) is called the *top bar* and includes the Activities menu (C), the day and time and a collection of buttons and icons including network status, audio volume, battery power and other status and account settings. The application menu for the currently active application running on the desktop will also appear in the top bar. Figure 7-2, for example, shows the application menu for the Terminal program:

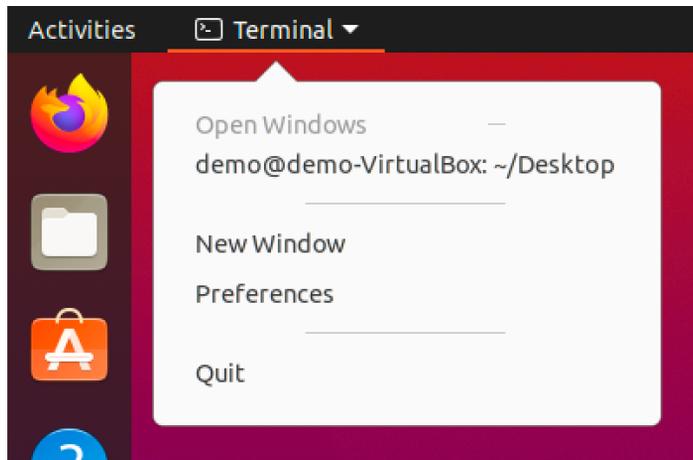


Figure 7-2

7.3 Launching Activities

Applications and utilities are launched using the Activities overview dashboard (referred to as the *dash*) which may be displayed either by clicking on the Activities button in the top bar or pressing the *special key* on the keyboard. On Windows keyboards this is the Windows key, on macOS the Command key and on Chromebooks the key displaying a magnifying glass.

When displayed, the dash will appear as shown in Figure 7-3 below:

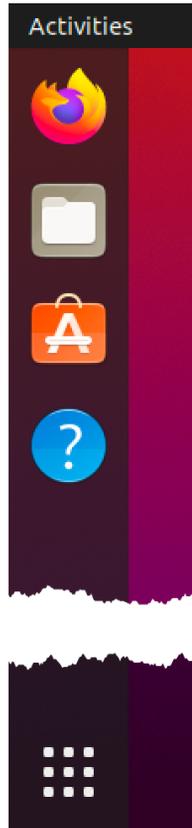


Figure 7-3

By default the dash will display an icon for a predefined set of commonly used applications and will also include an icon for any applications that are currently running. If the application is currently running it will appear with a dot marker to the left of the icon and if multiple copies are running a dot will appear for each instance.

To launch an application, simply click on the icon in the dash.

To find an application not included on the dash, one option is to select the bottom most icon (the square comprising nine dots) to display a browsable list of applications as shown in Figure 7-4:

A Guided Tour of the GNOME 3 Desktop

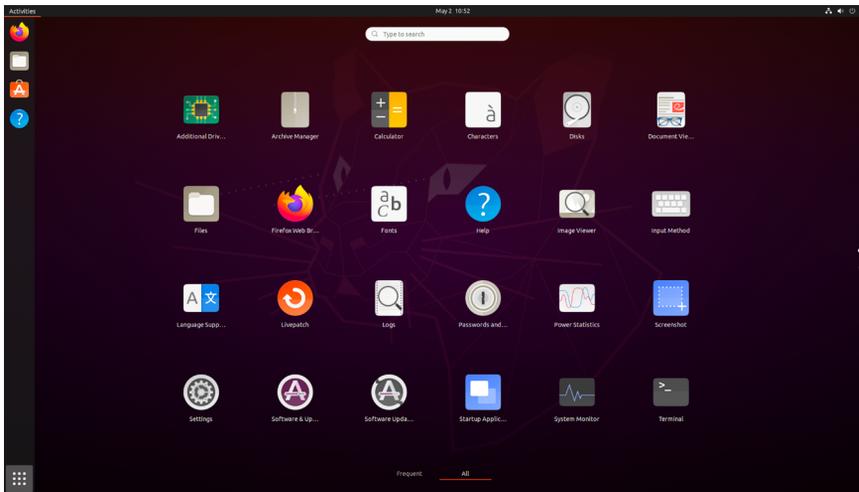


Figure 7-4

Note that the list can be filtered to display all applications or only those used frequently by selecting the buttons at the bottom center of the screen. It is also important to be aware that some entries in the list are actually folders holding additional applications.

An alternative to browsing the applications is to perform a search using the search bar which appears when the Activities menu is clicked as shown in Figure 7-5:

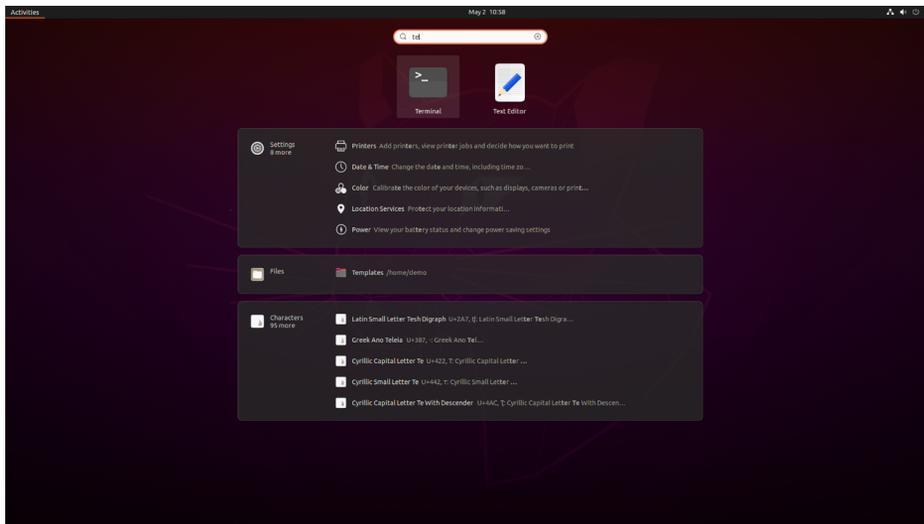


Figure 7-5

As text is typed into the search box, the list of possible matches will be refined.

To add an application to the dash for more convenient access, locate the icon for the application, right-click on it and select the *Add to Favorites* menu option:

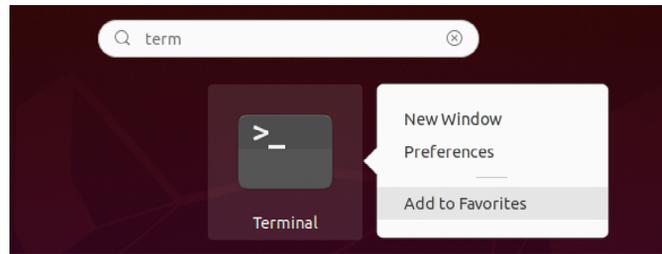


Figure 7-6

To remove an app from the dash, right-click on the icon in the dash and select *Remove from Favorites*.

7.4 Managing Windows

As with other desktop environments, applications run on GNOME in windows. When multiple application windows are open, the Super + Tab keyboard shortcut will display the switcher panel (Figure 7-7) allowing a different window to be chosen as the currently active window (the Super key is either the Windows key or, in the case of a Mac keyboard, the Cmd key):

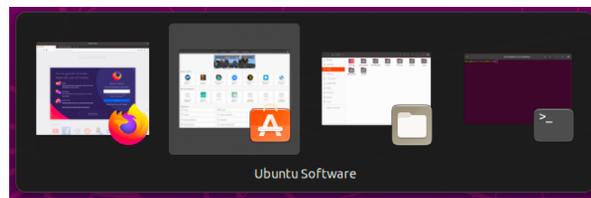


Figure 7-7

To cycle backwards through the icons in the switcher, use the Shift + Tab keyboard shortcut.

To maximize a window so that it fills the entire screen click the title bar and drag the window to the top of the screen. To return the window to its original size, click on the title bar and drag downwards. Alternatively, simply double-click on the title bar to toggle between window sizes. Similarly, dragging a window to the left or right side of the screen will cause the window to fill that half of the screen.

7.5 Using Workspaces

The area of the screen where the application windows appear is referred to as the *workspace* and GNOME 3 allows multiple workspaces to be configured. To create a new workspace, display the Activities overview and move the mouse pointer to the far right of the screen to display the workspaces panel:

A Guided Tour of the GNOME 3 Desktop

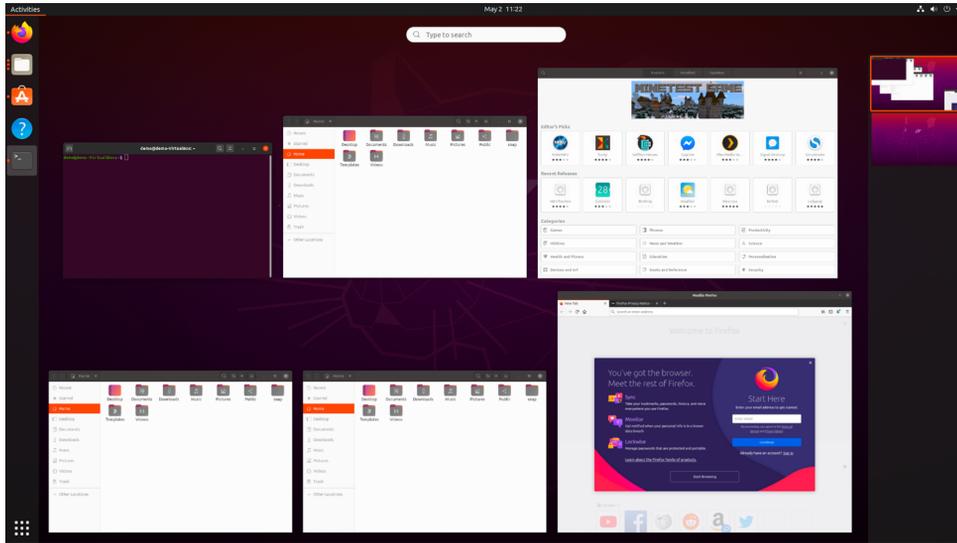


Figure 7-8

To switch to a different panel, simply select it from the list. To move a window from one workspace to another, display the workspaces panel and drag and drop the application window (either the actual window from the current workspace or the thumbnail window in the workspaces panel) onto the destination workspace. When a window is added to a blank workspace, another blank workspace is added to the workspace panel, allowing multiple workspaces to be created.

To remove a workspace either close all the windows on that workspace, or move them to another workspace.

7.6 Calendar and Notifications

When the system needs to notify you of an event (such as the availability of system or application updates), a popup panel will appear at the top of the workspace. Access to the calendar and any previous notifications is available by clicking on the day and time in the top bar as shown in Figure 7-9:

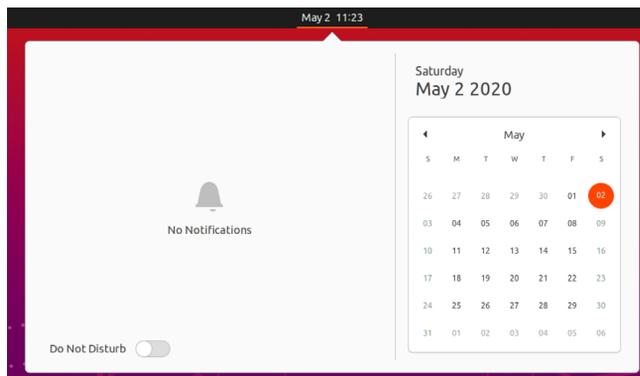


Figure 7-9

7.7 Desktop Settings

To access the Settings application, click on the down arrow on the far right of the top bar and select the button with the tools icon as highlighted in Figure 7-10:

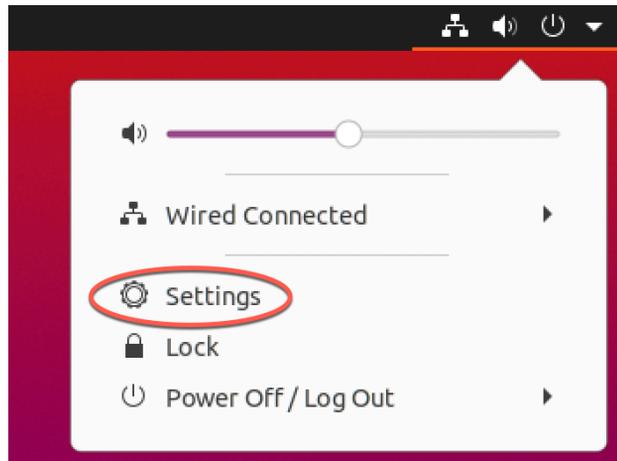


Figure 7-10

The Settings application provides a wide range of options such as Ethernet and WiFi connections, screen background customization options, screen locking and power management controls and language preferences. To explore the settings available in each category, simply select an option from the left-hand panel in the Settings window:

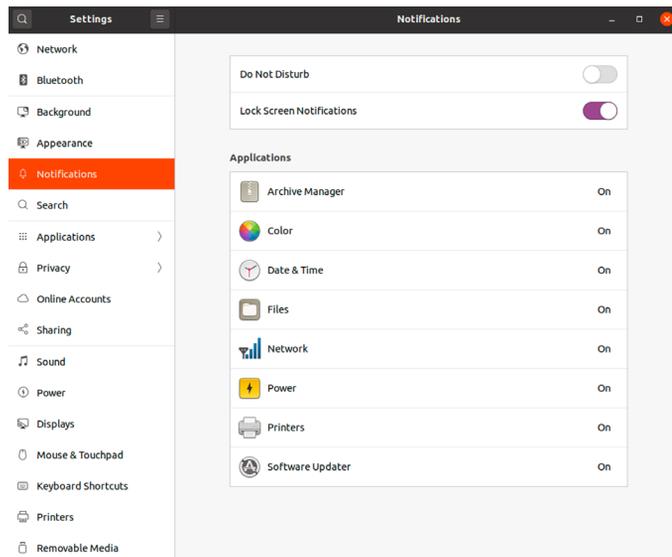


Figure 7-11

The menu shown in Figure 7-10 above also includes options to switch user, adjust audio volume, change to a different WiFi network and to log out, restart or power off the system.

7.8 Customizing the Dash

The size, position and behavior of the dash (also referred to as the dock) may be changed from within the Settings app by selecting the Appearance option as shown in Figure 7-12:

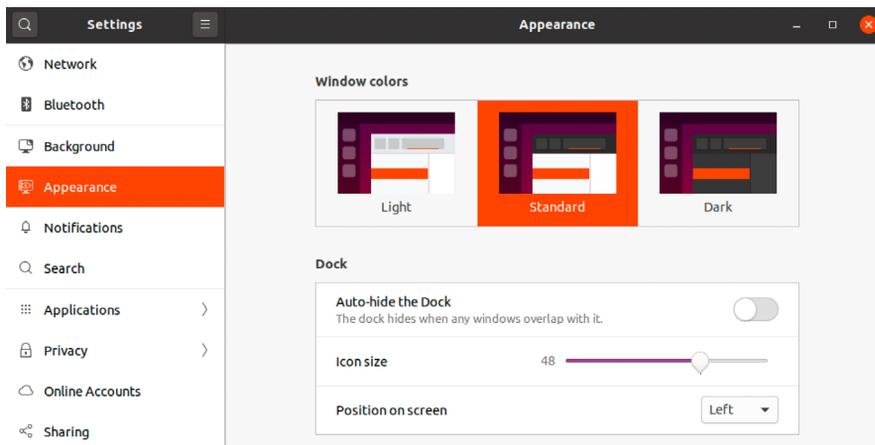


Figure 7-12

The position can be set to any of the four sides of the screen while the Icon size can be used to reduce the size of the dock. Finally, the auto-hide dock option, if enabled, will cause the dock to recede from view until the mouse pointer moves to the edge of the screen where it is currently located.

7.9 Switching to Dark Mode

In addition to providing dock settings, the Appearance panel of the Settings app also allows the desktop to be switched between light, standard and dark modes. Figure 7-13, for example, shows dark mode selected:

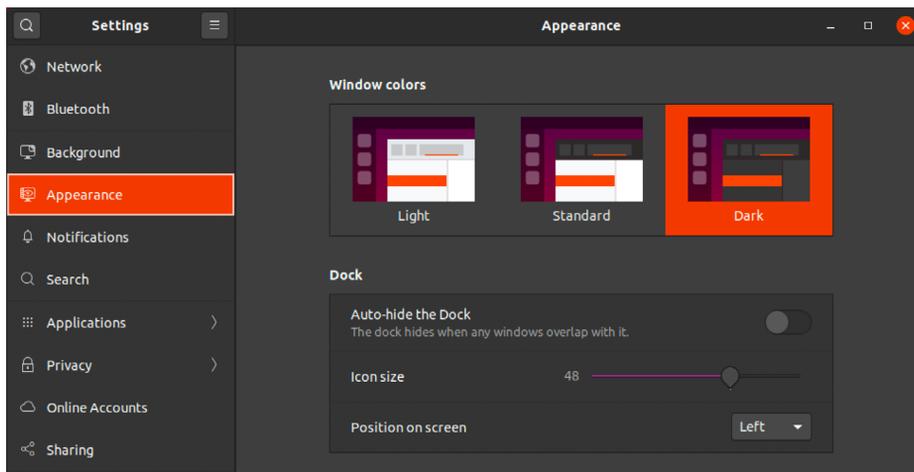


Figure 7-13

7.10 Installing Ubuntu Software

In common with other operating systems such as macOS, Windows, iOS and Android, Ubuntu has an “app store” in the form of the Ubuntu Software tool. An icon is usually placed in the dash as indicated in Figure 7-14:



Figure 7-14

Once loaded, the tool provides a list of applications available for installation on the system grouped together based on categories, recommendations and editor’s picks. The library of available applications may also be searched to find a specific item:

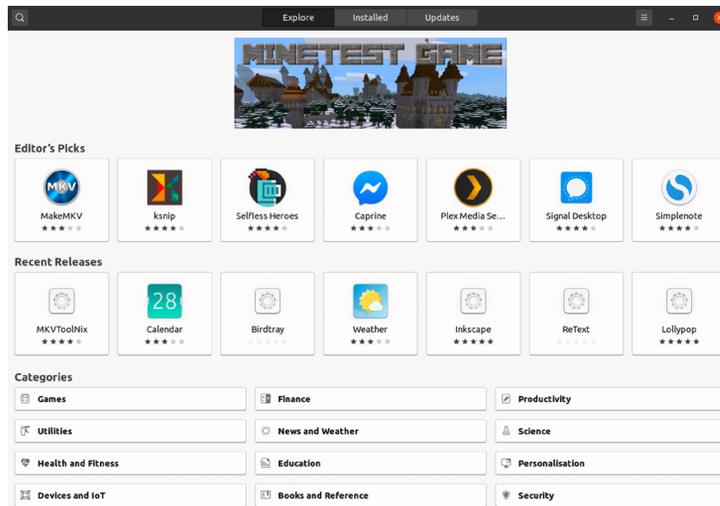


Figure 7-15

To install an application, simply select it in the Ubuntu Software window and click on the *Install*

A Guided Tour of the GNOME 3 Desktop

button:

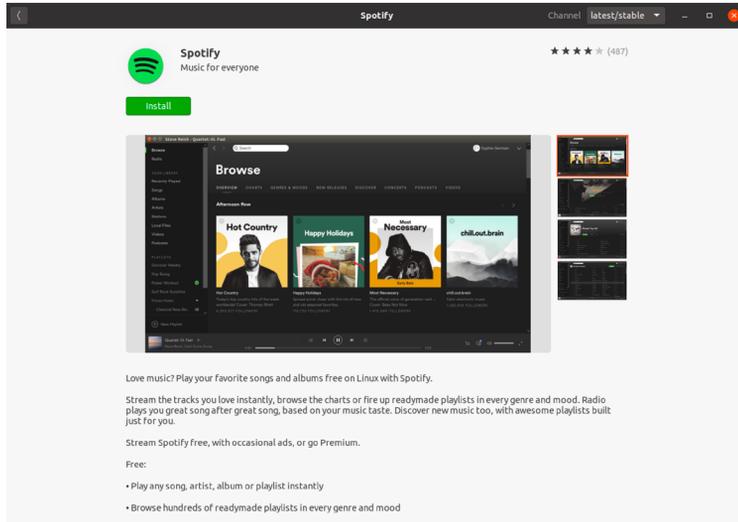


Figure 7-16

7.11 Beyond Basic Customization

The GNOME 3 desktop is, by design, a clean and uncluttered environment with minimal customization options. That does not mean, however, that it is not possible to make additional changes to the desktop. In fact, the GNOME Project has developed a tool called GNOME Tweaks for this very purpose. Use the following commands to install and run this tool:

```
# apt install gnome-tweaks  
$ gnome-tweaks
```

Once GNOME Tweaks has loaded, the interface shown in Figure 7-17 will appear:

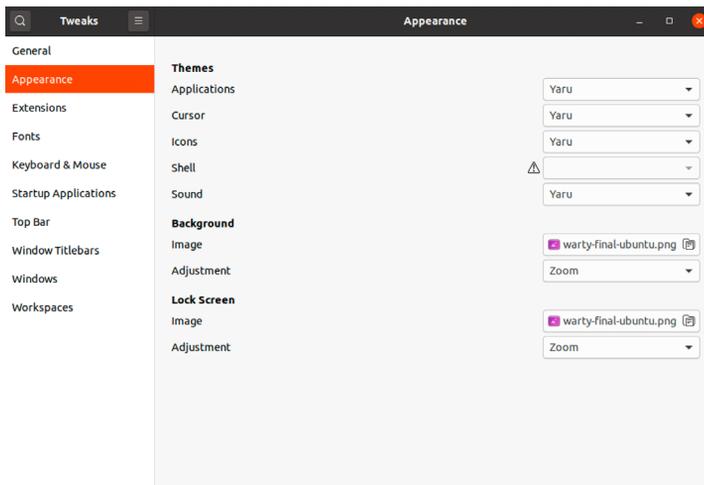


Figure 7-17

A wide range of options for customizing the desktop are now available. Too many, in fact, to cover in this chapter so take some time to experiment with these settings before proceeding to the next chapter.

7.12 Summary

Ubuntu includes the GNOME 3 desktop environment which may either be included during the initial installation or installed later using the *tasksel* command-line tool. Unlike most other desktop environments, GNOME 3 is intended to provide a clean and easy to use windowing user interface. Key areas of the GNOME 3 desktop include the top bar, Activities overview and dash. In addition, GNOME 3 supports multiple workspaces keeping running applications organized and the screen uncluttered. A variety of configuration options is also available within the Settings app including desktop background settings, audio, network configuration and WiFi network selection.

